Assessment of predictive relevance of covariates in Gaussian process models

Topi Paananen

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 31.1.2018

Supervisor and advisor

Prof. Aki Vehtari

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Topi Paananen

Title Assessment of predictive relevance of covariates in Gaussian process models

Degree programme Master's Programme in Engineering Physics

Major Engineering Physics **Code of major** SCI3056

Supervisor and advisor Prof. Aki Vehtari

Date 31.1.2018 **Number of pages** 57 **Language** English

Abstract
The thesis introduces two novel covariate selection methods for Gaussian process models. The methods sort the covariates of a full Gaussian process model based on predictive relevance by examining the posterior predictive distribution in the vicinity of the training points. Experiments conducted on synthetic and real world data sets demonstrate improved variable selection compared to automatic relevance determination, a commonly used existing method. The new methods are shown to be more consistent and produce submodels with a better predictive performance. The proposed methods are expected to be useful in simplifying and interpreting complex Gaussian process models.

Keywords Gaussian process, covariate selection, Bayesian inference

Aalto University
School of Science

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Topi Paananen

Työn nimi Kovariaattien merkitsevyyden vertailu Gaussisten prosessien avulla rakennetuissa tilastollisissa malleissa

Koulutusohjelma Master's Programme in Engineering Physics

Pääaine Engineering Physics **Pääaineen koodi** SCI3056

Työn valvoja ja ohjaaja Prof. Aki Vehtari

Päivämäärä 31.1.2018 **Sivumäärä** 57 **Kieli** Englanti

Tiivistelmä

Tässä työssä esitetään kaksi uutta muuttujanvalintamenetelmää Gaussisten prosessien avulla rakennetuille tilastollisille malleille. Menetelmät järjestävät kovariaatit perustuen niiden kykyyn ennustaa selitettävän muuttujan arvoja. Tämä tehdään tarkastelemalla täyden mallin tuottamia ennusteita lähellä mallin sovitukseen käytettyjä datapisteitä. Kovariaattien järjestyksen perusteella voidaan rakentaa yksinkertaisempi malli käyttämällä vain parhaita muuttujia. Menetelmien kykyä järjestää kovariaatit niiden merkitsevyyden mukaan tutkittiin simuloitujen sekä avointen tietoaineistojen avulla rakennetuissa muuttujanvalintaongelmissa. Tulokset osoittavat, että uudet menetelmät järjestävät muuttujat johdonmukaisemmin kuin yleisesti käytetty olemassaoleva ARD-menetelmä, sekä valitut muuttujat ennustavat selitettävää muuttujaa paremmin. Esiteltyjen menetelmien uskotaan olevan hyödyksi yksinkertaistamaan ja tulkitsemaan monimutkaisia Gaussisten prosessien avulla rakennettuja malleja.

Avainsanat Gaussinen prosessi, selittävien muuttujien valinta, Bayesilainen tilastotiede

Preface

This work was carried out in the Probabilistic Machine Learning group in the Department of Computer Science at the Aalto University.

I want to thank my advisor and supervisor Aki Vehtari for giving me the opportunity to carry out this work and for guiding me throughout the process. I also want to thank my colleagues for all the fruitful discussions and the inspiring working environment. I especially thank Juho Piironen and Michael Riis Andersen who have helped me in various ways during the thesis work.

Espoo, January 31, 2018

Topi Paananen

Contents

Symbols and abbreviations

Abbreviations

ARD	Automatic relevance determination
BMA	Bayesian model averaging
CV	Cross-validation
GP	Gaussian process
HMC	Hamiltonian Monte Carlo
KL	Kullback-Leibler
LOO-CV	Leave-one-out cross-validation
MAP	Maximum a posteriori
MCMC	Markov chain Monte Carlo
ML	Maximum likelihood
MLPD	Mean log-predictive density
NUTS	No-U-turn sampler
SE	Squared exponential

General notation

a, b, c	Scalars
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	Column vectors
a_i	i'th element of vector $\mathbf{a}$
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	Matrices
$\mathbf{A}^\intercal$	Matrix transpose
$\mathbf{A}^{-1}$	Matrix inverse
$\mathbf{a}_i$	i'th row of matrix $\mathbf{A}$
$a_{i,j}$	j'th element of the i'th row of matrix $\mathbf{A}$
$\mathbf{I}$	Identity matrix

Operators

$\mathrm{Cov}[\cdot, \cdot]$	Covariance
$D_{\mathrm{KL}}(p \,\|\| \, q)$	Kullback-Leibler divergence from distribution p to distribution q
$\mathbb{E}[\cdot]$	Expected value
$\mathrm{Var}[\cdot]$	Variance

Symbols

$\mathcal{M}$	Model structure and other modelling assumptions
$\mathcal{D}$	Observed dataset of n observations, $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$
$\mathbf{x}^*$	Unobserved input variable
y^*	Unobserved output variable
n	Number of observed examples
m	Number of covariates
$\mathbf{w}$	Vector of parameters
$\boldsymbol{\theta}$	Vector of hyperparameters

1 Introduction

In recent years, deep and complex machine learning models have gained a large amount of attention due to leaps made in performance in a variety of applications, such as object classification, speech recognition, and recommender systems. However, the theoretical understanding of many complex models is still incomplete, and the results produced by these models are not fully understood. A great deal of research still needs to be done in order to learn interpreting the decisions made by these models. This is especially important for safety-critical applications, such as self-driving cars, where the decisions of the machine learning model may crucially affect the lives of humans.

In practice, it is difficult to know which input variables are needed to solve a certain problem or predict a certain target variable. While it may be feasible to include every plausible covariate in a model, analyzing such a model is often difficult. Covariate selection is a widely studied problem in machine learning that can be used to simplify statistical models in order to make them more interpretable. From the full set of covariates, some may be totally irrelevant for a given problem, and identifying and removing them is therefore beneficial. In addition, reducing the number of input variables often reduces measurement and modelling costs in the future.

This thesis studies covariate selection methods specific to Gaussian process models. Gaussian processes offer a flexible nonparametric framework for Bayesian regression. While they are suitable for predictive modelling in a variety of applications, interpreting the models may be difficult due to their flexible nature. This thesis shortly reviews covariate selection methods found in the literature, and those specific to Gaussian process models are introduced in more detail. As a background, the essential theory of Gaussian processes is introduced together with the basics of Bayesian inference.

The main contribution of this work is the introduction and implementation of two novel covariate selection methods for Gaussian process models. The theoretical foundation of the methods is explained, and they are motivated through the shortages of existing methods. The properties of these methods are further investigated in an array of numerical experiments with both simulated and real data sets. The experiments indicate that the new methods lead to models with improved predictive performance compared to previously used methods without increasing the computational complexity prohibitively. This is a result of several factors, one of which is the improved identification of linear but important covariates. The differences in predictive performance are explained with the theoretical properties of the different methods.

Some of the results in this thesis are presented in a shorter format in (Paananen et al., 2017), which is a part of the contributions of the thesis work. This thesis will present the methods slightly more thoroughly and provide more background for the discussed topics. In addition, the thesis includes additional numerical experiments. The experiments with simulated data sets are replicated with additional irrelevant covariates, and the estimation consistency results are shown in more detail for all of the real world data sets.

The thesis is structured as follows. The discussion begins in Section 2 with an introduction to several concepts related to Bayesian statistics. The principles of Bayesian inference are introduced together with methods to perform it in practice. Section 3 continues with a review of Bayesian model selection, which is an important concept for understanding the rest of this thesis. Section 4 presents the theory of Gaussian processes and their use as regression models. Section 5 introduces covariate selection as a special case of model selection, and reviews existing approaches to the problem in the Gaussian process framework. The introduced covariate selection methods serve as motivation for this work and for the new methods which are presented in Section 6. In Section 7 these two methods are compared to previous methods in several numerical experiments, which include constructed toy data sets as well as freely available benchmark data sets. Finally, Section 8 summarizes and discusses the key results of this thesis.

2 Bayesian modelling

2.1 Bayesian inference

Given some set of data $\mathcal{D}$, the basis for a Bayesian statistical analysis is a probabilistic model $\mathcal{M}$, which is a description that models uncertainty with probability (Gelman et al., 2014). The purpose of the model is to act as a hypothesis for the unknown process that generated the data. Often the model contains some parameters $\mathbf{w}$, which are treated as random variables as the true values are unknown. Bayesian inference refers to the process of fitting the model to available data, and expressing the result as a probability distribution of the unknown model parameters. The aspect that defines inference as Bayesian, is quantifying uncertainty about the parameters with probability, and transferring that uncertainty into uncertainty about future observations. Meeting the requirements of a full Bayesian treatment is often difficult in practice, but nevertheless, the framework provides useful guidelines for statistical inference.

Unless otherwise stated, this thesis will consider models that predict continuous values of a one-dimensional output variable y given p input variables $\mathbf{x} = \{x_1, \ldots, x_p\}$, which are called *regression models*. We will denote a sample of n data points as $\mathcal{D} = (\mathbf{X}, \mathbf{y})$, where $\mathbf{y} = [y^{(1)} \ldots y^{(n)}]^\mathsf{T}$ is the vector of output values and $\mathbf{X} = [\mathbf{x}^{(1)} \ldots \mathbf{x}^{(n)}]^\mathsf{T}$ is the design matrix. Before the data is observed, the predictive distribution of the observations $\mathbf{y}$ for given values of the parameters $\mathbf{w}$ is characterized by the *sampling distribution* $p(\mathbf{y}|\mathbf{w}, \mathcal{M})$. When the data is observed, the sampling distribution as a function of $\mathbf{w}$ is called the *likelihood*, stating how probable the data is given some $\mathbf{w}$. Prior beliefs about the unknown model parameters are determined by the *prior distribution* $p(\mathbf{w}|\mathcal{M})$.

Bayesian inference is the process of combining prior information with new information provided by the data into a posterior distribution $p(\mathbf{w}|\mathbf{y}, \mathcal{M})$. The posterior follows from the Bayes' rule, which states the conditional distribution of the model parameters $\mathbf{w}$ given available data $\mathbf{y}$ and the model assumptions as

$$p(\mathbf{w}|\mathbf{y}, \mathcal{M}) = \frac{p(\mathbf{y}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\mathcal{M})}{p(\mathbf{y}|\mathcal{M})} = \frac{p(\mathbf{y}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\mathcal{M})}{\int p(\mathbf{y}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\mathcal{M})\mathrm{d}\mathbf{w}}. \tag{1}$$

The posterior distribution contains all the information available about the parameters and their uncertainty. The denominator in the Bayes' theorem is called either *evidence* or *marginal likelihood*, and it integrates the likelihood over the prior information about the parameters. The marginal likelihood normalizes the posterior into a proper probability distribution.

The Bayes' rule provides an intuitive way of updating one's beliefs when new data is available, by using the previous posterior as the new prior distribution. Inference can thus be performed sequentially, and the effect of new data on the posterior distribution can be observed. Updating the posterior beliefs sequentially is consistent and produces the same end result as if all the data was used at once.

After observing data $\mathcal{D}$, the posterior distribution of the parameters, $p(\mathbf{w}|\mathbf{y}, \mathcal{M})$, can be straightforwardly used to model the uncertainty about an unobserved target value y^* at any input point $\mathbf{x}^*$. In the Bayesian framework, the *posterior predictive distribution* of y^* is given by averaging the conditional predictions of unobserved data over the posterior distribution of $\mathbf{w}$:

$$p(y^*|\mathbf{y}, \mathcal{M}) = \int p(y^*, \mathbf{w}|\mathbf{y}, \mathcal{M}) \mathrm{d}\mathbf{w} = \int p(y^*|\mathbf{w}, \mathcal{M}) p(\mathbf{w}|\mathbf{y}, \mathcal{M}) \mathrm{d}\mathbf{w}. \tag{2}$$

The above equation holds given the assumption that the future observation y^* is conditionally independent of $\mathbf{y}$ for particular values of $\mathbf{w}$.

2.2 Approximate inference

The integrals arising from Bayesian inference are analytically tractable only in simple modelling situations, and it is often necessary to resort to approximations. It is essential to be aware of the accuracy of each approximation, as inaccuracies can make the conclusions about the posterior to be invalid. Methods for approximating intractable integrals can be roughly categorized into distributional and numerical methods.

Distributional approximations aim to mimic the true posterior distribution with a similar but simpler distribution that allows the inference to be conducted more easily. The approximations may assume a particular parametric form or a certain factorization for the posterior, for example. By construction, distributional methods will never generate exact results, but they often have a rather low computational complexity.

Numerical integration methods aim to directly approximate an integral by computing the function values at a finite number of points. The methods can be categorized into deterministic and simulation methods based on whether the points are chosen in a deterministic or stochastic way. Deterministic methods are efficient at approximating low-dimensional integrals, but they do not scale well to higher dimensions. On the other hand, simulation methods are more efficient at approximating high-dimensional integrals, and they are thus commonly used in approximate Bayesian inference. For a good overview of approximate inference methods, see e.g. (Bishop, 2006, chap. 10).

In this thesis, simulation methods based on Markov chains are used, and they are briefly presented in the next section.

2.2.1 Markov chain Monte Carlo

An extremely popular class of simulation methods for approximate inference are Markov chain Monte Carlo (MCMC) methods. The methods are based on constructing a Markov chain that has the true probability distribution as its equilibrium distribution. MCMC methods approximate the integral exactly in the asymptotic limit of infinite computational resources. Even though they produce a set of dependent samples, careful tuning and monitoring of the generated chain can mitigate the problem and make them effective in practice.

The mechanism of MCMC methods is the following. If $\mathbf{w}$ is a random variable with a distribution $p(\mathbf{w})$, then the expectation of an arbitrary function $f(\mathbf{w})$ can be approximated by drawing N samples from $p(\mathbf{w})$ and averaging over the samples as

$$\mathbb{E}_p[f(\mathbf{w})] = \int f(\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w} \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{w}^{(i)}). \tag{3}$$

The typical application is Bayesian inference is the situation where p is the posterior distribution of some model parameters $\mathbf{w}$, and f is the predictive distribution conditioned on $\mathbf{w}$, giving rise to an estimate for the posterior predictive distribution:

$$p(y^*|\mathbf{y},\mathcal{M}) = \int p(y^*|\mathbf{w},\mathcal{M})p(\mathbf{w}|\mathbf{y},\mathcal{M})\mathrm{d}\mathbf{w} \approx \frac{1}{N}\sum_{i=1}^{N} p(y^*|\mathbf{w}^{(i)},\mathcal{M}). \tag{4}$$

In this thesis, an MCMC algorithm called *Hamiltonian Monte Carlo* (HMC) (Duane et al., 1987) is utilized for integrating over intractable integrals in Gaussian process models. HMC introduces Hamiltonian dynamics in order to achieve more efficient proposals and avoid random walk behaviour. This is done by augmenting the parameter space with auxiliary momentum variables and simulating the system with Hamiltonian equations. Hamiltonian Monte Carlo can potentially make sampling significantly more efficient, but it requires the gradients of the distribution to be tractable. In addition, it introduces several parameters that need to be tuned, making it more difficult to use compared to simpler MCMC algorithms. However, automatically adapting methods have been developed recently, such as the no-U-turn sampler (NUTS) (Hoffman and Gelman, 2014; Betancourt, 2016).

2.3 Bayesian decision theory

The posterior distribution $p(\mathbf{w}|\mathbf{y}, \mathcal{M})$ summarizes all the available information about $\mathbf{w}$ given the data and the model assumptions. Therefore, it can be used to evaluate the probability of any statement regarding the parameters. A simple example is to assess the probability of a point estimate $\hat{\mathbf{w}}$ for the true values of the unknown parameters $\mathbf{w}$. A point estimate might be used, for example, when integrating over the full posterior distribution is computationally too expensive. While using a point estimate is fundamentally not Bayesian, it is not uncommon to have to resort to such an approximation because of limited computational resources.

The difference between the true parameter value and the point estimate can be quantified with the expected loss over the posterior distribution

$$R_L(\hat{\mathbf{w}}) = \int L(\hat{\mathbf{w}}, \mathbf{w}) p(\mathbf{w}|\mathbf{y}, \mathcal{M}) \mathrm{d}\mathbf{w}, \tag{5}$$

where $L(\hat{\mathbf{w}}, \mathbf{w})$ is a loss function that determines the error made when estimating $\mathbf{w}$ with $\hat{\mathbf{w}}$. The optimal Bayesian point estimate is one that minimizes the expected loss for a given loss function. Two common loss functions are the absolute error $L(\hat{\mathbf{w}}, \mathbf{w}) = ||\hat{\mathbf{w}} - \mathbf{w}||_1$ and the squared error $L(\hat{\mathbf{w}}, \mathbf{w}) = ||\hat{\mathbf{w}} - \mathbf{w}||_2^2$, which are minimized by the posterior median and mean of $\mathbf{w}$, respectively.

The mode of the posterior distribution is called the *maximum a posteriori* (MAP) estimate, and it corresponds to a loss function that is zero in the case of exact estimation, and a positive constant otherwise. The MAP estimate is often easy to compute because it can be found via optimization without normalizing the posterior distribution:

$$\hat{\mathbf{w}}_{\mathrm{MAP}} = \arg\max_{\mathbf{w}} \, p(\mathbf{w}|\mathbf{y}, \mathcal{M}) = \arg\max_{\mathbf{w}} \, p(\mathbf{y}|\mathbf{w}, \mathcal{M}) p(\mathbf{w}|\mathcal{M}). \tag{6}$$

If the prior $p(\mathbf{w}|\mathcal{M})$ is uniform over $\mathbf{w}$, the MAP estimate is equal to the mode of the likelihood, which is referred to as the maximum likelihood (ML) estimate. Because MAP estimation incorporates the prior, it is sometimes called penalized or regularized maximum likelihood estimation. For high-dimensional distributions, the MAP point estimate is problematic, because it maximizes the posterior density instead of the posterior mass. Because of an effect called concentration of measure, the mode of a high-dimensional probability distribution can be located far from the posterior mass. Another drawback with the MAP estimate is that it is not invariant to nonlinear transformations of the parameters.

2.4 Hierarchical models

Many statistical problems can be effectively modelled by establishing connections between multiple model parameters. Dependencies between the parameters can be created with a joint probability model, and the principle of updating one's beliefs using the Bayes' rule can be extended to such hierarchically specified models (Gelman et al., 2014). While this thesis does not study hierarchical models as such, the terms and concepts introduced here apply perfectly to Gaussian process models.

Consider a two-layer hierarchical model, where observations can be divided into groups so that observations $(\mathbf{X}^{(j)}, \mathbf{y}^{(j)}) \subseteq (\mathbf{X}, \mathbf{y})$ are controlled by parameters $\mathbf{w}_j \subseteq \mathbf{w}$. The groups of parameters $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_k\}$ then constitute the first layer of the hierarchical model. In a hierarchical model, it is assumed that the parameters of the first layer $\{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_k\}$ have a common population distribution, which is controlled by some additional parameters $\boldsymbol{\theta}$. The parameters $\boldsymbol{\theta}$ constitute the second layer of the hierarchical model and are often called *hyperparameters*, and their prior distribution is subsequently called a *hyper-prior*. For a two-layer hierarchical model, the posterior distribution of $\mathbf{w}$ is given as the product of the likelihood and the prior

$$p(\mathbf{w}|\mathbf{y}, \boldsymbol{\theta}, \mathcal{M}) = \frac{p(\mathbf{y}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{M})}{p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})} = \frac{p(\mathbf{y}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{M})}{\int p(\mathbf{y}|\mathbf{w}, \mathcal{M})p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{M})\mathrm{d}\mathbf{w}}. \tag{7}$$

The marginal likelihood of the bottom layer, $p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})$, plays the role of the likelihood at the second layer, and the posterior over the hyperparameters is

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M}) = \frac{p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})}{p(\mathbf{y}|\mathcal{M})} = \frac{p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})}{\int p(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})\mathrm{d}\boldsymbol{\theta}}. \tag{8}$$

Because the posterior of $\mathbf{w}$ now depends on $\boldsymbol{\theta}$, the posterior predictive distribution must integrate over both (7) and (8). In principle, the hierarchical structure can be extended even further than two layers.

2.5 Modelling perspectives

In practical statistical modelling problems, the true model that generated the data is unknown. The need to construct an approximative model for describing a particular problem is therefore essential in statistical inference. Different perspectives for relating to the true data generating model were proposed by Bernardo and Smith (1994), who introduced three different modelling views: $\mathcal{M}$-closed, $\mathcal{M}$-completed, and $\mathcal{M}$-open. The $\mathcal{M}$-closed view is the most restricted, and it assumes that one has a set of candidate models, one of which is the true model. The $\mathcal{M}$-completed view

instead considers a reference model, which is considered as the best approximation to the true data generating process without assuming that it is fully true. The $\mathcal{M}$-open view is the loosest, and it makes minimal assumptions about the true model. The different views about the true data generating model give rise to different methods for constructing the belief model. However, the modelling perspectives should not always be interpreted strictly, as some approaches cannot be categorized into any single view, and some approaches may combine properties from different views.

In the equations (1)-(8), every term has been conditioned on the fixed modelling assumptions. However, there is no special reason to keep the model fixed if one is uncertain about the true model, and with the Bayesian framework it is possible to account for this uncertainty. By specifying a set of candidate models $\{\mathcal{M}_i\}_{i=1}^K$ as the model space, the posterior distribution of the models $\{\mathcal{M}_i\}_{i=1}^K$ is given by the Bayes' rule as

$$p(\mathcal{M}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{M})p(\mathcal{M})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathcal{M})p(\mathcal{M})}{\sum_{i=1}^K p(\mathbf{y}|\mathcal{M}_i)p(\mathcal{M}_i)}. \tag{9}$$

$p(\mathcal{M}|\mathbf{y})$ is a discrete probability distribution that determines the posterior probability of each model in the model space, given a prior $p(\mathcal{M})$ and the data. In this context, $p(\mathbf{y}|\mathcal{M})$ is sometimes called the *model evidence*, but it is simply the marginal likelihood from the denominator of equations (1) and (8).

The key benefit of computing the posterior probabilities of different models is that the uncertainty about the correct model can be transferred to the uncertainty about predictions of the target variable. Summing the products of the posterior predictive distributions and the model evidences of each model yields the *Bayesian model averaging* (BMA) predictive distribution

$$p(y_*|\mathbf{y}) = \sum_{i=1}^K p(y_*|\mathbf{y}, \mathcal{M}_i)p(\mathcal{M}_i|\mathbf{y}). \tag{10}$$

The key difference to equation (2) is that the distribution is no longer conditioned on a fixed model, but instead averages over all the models according to their relative probabilities. Predicting with BMA has demonstrated to perform well in practice, and it also has a sound theoretical justification (Raftery and Zheng, 2003; Hoeting et al., 1999). However, it may lead to poor results if the model space is not appropriate for the problem, because models outside the space are not considered.

3 Model selection

Model selection is a general term that can refer to many decision problems encountered when constructing a statistical model. The decisions are typically discrete choices between types of models or between covariates to include in the model. Speaking more loosely, model selection can also refer to, for example, choosing point estimates for some hyperparameters if one wants to reduce the computational cost compared to a Bayesian treatment. Model selection can therefore be useful for solving practical modelling challenges, even though restricting the model will ignore some uncertainties.

The goal of model selection is to obtain a model that agrees with the process that generated the data for which the model is constructed. This includes agreeing with both seen observations and unseen future observations. The utility of a model is most often quantified by its capability to predict new observations. This thesis focuses on *predictive model selection*, which refers to the assessment of predictive performance of models. This section will introduce some concepts and methods for model selection. For a more thorough discussion of model selection, see e.g. (Vehtari and Ojanen, 2012; Piironen and Vehtari, 2017).

Model selection is often motivated through a concept called the *bias-variance trade-off*, which is an inescapable dilemma in supervised learning. The dilemma arises from the aim to generalize the model to unseen data when only a finite amount of training data is available. It is typically impossible to identify all relevant regularities from the training data while maintaining the generalization ability. A highly complex model can capture more structure from the data, but may have a tendency to confuse noise with structure. Conversely, a simple model will make erroneous assumptions and miss some structure in the data, but will avoid learning the noise from the training sample. Therefore, a model that can balance between these two extremities will be useful in practical modelling problems.

3.1 Predictive performance

The predictive performance of a statistical model can be quantified with a utility function, which evaluates the model by comparing its predictions to observations and assigns a reward to accurate predictions. Depending on the context, it may be more appropriate to define a loss function instead of a utility function. The principle of both is exactly the same, and they can be interchanged with a reversal of sign. Here, we will stick to the convention of a utility function u, defined as a function that maps each prediction y^* of a model $\mathcal{M}$ into a scalar utility for a given unobserved value $\tilde{y}$. The closer the predictions of a candidate model are to the true state of the world,

the greater the generated utility score is. The goal of predictive model selection is to maximize the utility score.

If the goal is to find an optimal point prediction for future observations, a widely used utility function depends on the squared difference as

$$u_{\mathrm{SE}}(\mathcal{M}, \tilde{y}) = -(y^* - \tilde{y})^2, \tag{11}$$

which is maximized in the case of an exact prediction, $y^* = \tilde{y}$. The widespread use of the squared difference is largely based on its mathematical convenience. In the Bayesian framework, predictions are given as a posterior predictive distribution $p(y^* \,|\, \mathbf{y}, \mathcal{M})$ instead of a single point. Consequently, a widely used utility function is the density of the predictive distribution at the observation $\tilde{y}$, given by the logarithmic score

$$u_{\log}(\mathcal{M}, \tilde{y}) = \log p(y^* = \tilde{y} \,|\, \mathbf{y}, \mathcal{M}). \tag{12}$$

In order to choose a model that optimally predicts values for new observations, one needs to evaluate the utility function which depends on those observations. However, the utility function cannot be evaluated before the unknown future observations $\tilde{y}$ are actually observed. Therefore one often wants to maximize the *generalization utility*

$$\overline{u}_t(\mathcal{M}) = \mathbb{E}[u(\mathcal{M}, \tilde{y})] = \int p_t(\tilde{y}) u(\mathcal{M}, \tilde{y}) \mathrm{d}\tilde{y}, \tag{13}$$

where $p_t(\tilde{y})$ is the true probability distribution of new observations $\tilde{y}$. As the true distribution $p_t(\tilde{y})$ is also unknown, the generalization utility needs to be estimated.

The generalization utility has an important connection to information theory, because maximizing it with the logarithmic score equals minimizing the Kullback-Leibler (KL) divergence from the predictive distribution of the true model $\mathcal{M}_t$ to the used model $\mathcal{M}$. This arises from the fact that $\overline{u}_t(\mathcal{M})$ can be decomposed as

$$\overline{u}_t(\mathcal{M}) = \overline{u}_t(\mathcal{M}_t) - D_{\mathrm{KL}}(p_t(\tilde{y}) \,||\, p(\tilde{y} \,|\, \mathbf{y}, \mathcal{M})). \tag{14}$$

The Kullback-Leibler divergence from probability distribution $p(\mathbf{x})$ to distribution $q(\mathbf{x})$ is defined as (Kullback and Leibler, 1951)

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(\mathbf{x}) \log \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) \mathrm{d}\mathbf{x}. \tag{15}$$

This is a nonnegative and asymmetric quantity that measures how much the distribution $q(\mathbf{x})$ diverges from $p(\mathbf{x})$. More specifically, the KL divergence is the loss in Shannon information when using distribution q instead of the true distribution

p (Shannon, 1948).

3.2 Utility estimation

In practice, even the generalization utility in equation (13) cannot be computed because the true data generating distribution $p_t(\tilde{y})$ is unknown. This section will present methods for estimating the generalization utility, which provide the basis for predictive model selection. We will consider all of the methods by using the logarithmic score utility from equation (12).

A naive way for estimating the generalization utility of a model is to use the full available data set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ and estimate (13) by summing over the data set

$$\overline{u}_{\text{tr}} = \frac{1}{n} \sum_{i=1}^{n} \log p(y^* = y^{(i)} \,|\, \mathbf{x}^{(i)}, \mathcal{D}, \mathcal{M}). \tag{16}$$

In this way, the performance of the model is evaluated using the training data, and the estimator is called the training utility. However, this is a poor estimator of predictive performance, because a model that over-fits, i.e. fits to the noise in the training data, will produce a large training utility but will predict new observations poorly.

In order to avoid using the same data for training the model and assessing its performance, the data can be split into separate training and test data sets $\mathcal{D} = \{\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{ts}}\}$. If the model is trained with the training set and evaluated with the test set, the estimator is called the hold-out utility

$$\overline{u}_{\text{hold-out}} = \frac{1}{n_{\text{ts}}} \sum_{i \in I_{\text{ts}}} \log p(y^* = y^{(i)} \,|\, \mathbf{x}^{(i)}, \mathcal{D}_{\text{tr}}, \mathcal{M}), \tag{17}$$

where I_{ts} denotes the indices of points in the test set $\mathcal{D}_{\text{ts}}$ and n_{ts} is the size of the test set. Conversely to the training utility, the hold-out utility typically underestimates the generalization performance of the model, because using all the data for training would yield a more accurate model.

The choice of the sizes of the training and test data sets in the hold-out method is a trade-off between the predictive performance of the model and the variance in the utility estimate. A small test set yields a more accurate model but leads to high variance in the utility estimator. The accuracy of utility estimation can be improved with k-fold cross-validation (k-fold-CV). It extends the idea of hold-out such that the data is divided into k subsets, and each subset is alternately left for model validation

while all the other $k-1$ sets form the training set:

$$\overline{u}_{k\text{-CV}} = \frac{1}{n}\sum_{i=1}^{n} \log p(y^* = y^{(i)} \,|\, \mathbf{x}^{(i)}, \mathcal{D}_{\setminus I(i)}, \mathcal{M}), \tag{18}$$

where the test indices $I(i)$ have been removed from the training data, which is now $\mathcal{D}_{\setminus I(i)}$. Cross-validation provides a utility estimate for the whole data set, resulting in smaller variance, but requires that the model is trained k times. An extreme special case of k-fold-CV is obtained by setting $k = n$, i.e. leaving only one point for model validation and training the model with the rest, and repeating this n times. This procedure is called leave-one-out cross-validation (LOO-CV):

$$\overline{u}_{\text{LOO-CV}} = \frac{1}{n}\sum_{i=1}^{n} \log p(y^* = y^{(i)} \,|\, \mathbf{x}^{(i)}, \mathcal{D}_{\setminus i}, \mathcal{M}). \tag{19}$$

Cross-validation is a prevalent model selection method that avoids using the same data for model training and evaluation, but still enables computing the utility estimate from the whole data set. Its drawback is the relatively large computational cost, especially with LOO-CV. The amount of splits used is a trade-off between computational complexity and bias induced due to the incomplete training data.

3.3 Selection by evidence maximization

As outlined in Section 2, the Bayesian framework can naturally quantify uncertainty with probability at the different levels of modelling, including the parameters, hyperparameters, and model structure. This section describes model selection methods based on evidence maximization. Consider a situation where a statistician has a set of models that are deemed plausible for modelling a given problem. In order to simplify the modelling, the statistician may use only one of the models, and pick the one that has the greatest posterior probability. This is called the maximum a posteriori model

$$\mathcal{M}_{\text{MAP}} = \arg\max_{\mathcal{M}} p(\mathcal{M} \,|\, \mathbf{y}) = \arg\max_{\mathcal{M}} p(\mathbf{y} \,|\, \mathcal{M}) p(\mathcal{M}). \tag{20}$$

If the prior probabilities for each model are equal, the MAP model corresponds to the maximum of the marginal likelihood $p(\mathbf{y} \,|\, \mathcal{M})$.

Maximizing the marginal likelihood is a common model selection procedure, and it can naturally be used to also select point estimates for hyperparameters in a hierarchical model. The method is called *empirical Bayes*, *evidence framework*, or *type II maximum likelihood* (MacKay, 1992; Bernardo and Smith, 1994; Berger,

2013). The popularity of maximum marginal likelihood arises from the fact that it conforms to the principle of *Occam's razor* (Jefferys and Berger, 1992; Rasmussen and Ghahramani, 2001). This is a general principle stating that out of equally good hypotheses, the simplest should be chosen. In other words, the method intrinsically embodies a trade-off between the complexity of the model and the fit to data. The downside of the method is that optimization opens the possibility to severely over-fit the model.

4 Gaussian process models

This section introduces Gaussian processes and their application to Bayesian modelling. Gaussian process models are an important nonparametric class of models because of their convenient properties resulting from the Gaussian distribution. In the machine learning community, Gaussian process models have become a common approach for modelling nonlinear relationships between a target variable and a set of covariates. This section will introduce how Gaussian process models can be used for regression tasks, and how their properties can be controlled with the covariance function of the process.

4.1 Gaussian processes

A Gaussian process (GP) is an infinite collection of random variables, for which any finite subset $\{f^{(1)}, f^{(2)}, \ldots, f^{(n)}\} = \{f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \ldots, f(\mathbf{x}^{(n)})\}$ form a joint Gaussian distribution (Rasmussen and Williams, 2006)

$$p(f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \ldots, f(\mathbf{x}^{(n)})) = p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}). \tag{21}$$

The mean $\mathbf{m}$ and covariance $\mathbf{K}$ of the distribution are specified by the mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ of the Gaussian process $f(\mathbf{x})$:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \end{aligned} \tag{22}$$

The mean and covariance function completely specify the Gaussian process, which is denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{23}$$

Gaussian processes satisfy a consistency property, which means that if a particular GP specifies a joint distribution between two sets of values as $(\mathbf{f}^{(1)}, \mathbf{f}^{(2)}) \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$, then it also specifies $\mathbf{f}^{(1)} \sim \mathcal{N}(\mathbf{m}_1, \mathbf{K}_{11})$ and $\mathbf{f}^{(2)} \sim \mathcal{N}(\mathbf{m}_2, \mathbf{K}_{22})$, where $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{K}_{11}$, and $\mathbf{K}_{22}$ are subvectors or submatrices of $\mathbf{m}$ and $\mathbf{K}$. The consistency property implies that examining an additional set of variables $\mathbf{f}^{(2)}$ from the GP does not alter the distribution of the previously examined variables $\mathbf{f}^{(1)}$.

Gaussian process models utilize the GP framework by constructing a model where a particular Gaussian process is set directly as a prior over functions that map the inputs to the output(s). The chosen GP prior, specifically its mean and covariance function, represent the beliefs about what kind of functions are expected to model the

data. Contrary to parametric models, in the Gaussian process framework, inference is conducted directly about the functions without explicitly parametrizing them. For this reason, Gaussian process models are sometimes called nonparametric, as the GP prior corresponds to an infinite-dimensional parameter space.

In Gaussian process regression, the mean function is very often set to zero, and this convention is also adopted in this thesis. This is a convenient choice and agrees with prior beliefs if the output data is transformed to have zero mean during pre-processing. Moreover, this does not restrict the posterior process to have a zero mean. The Gaussian process framework can straightforwardly handle also a nonzero mean function $m(\mathbf{x})$ and make inference about its parameters. For example, inference over the parameters $\boldsymbol{\beta}$ of a linear mean function $m(\mathbf{x}) = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}$ can be conducted analytically (O'Hagan, 1978). However, the prior assumption of a linear trend in the latent function can also be modelled with a zero mean GP if a linear term $\mathbf{x}^\mathsf{T}\mathbf{x}'$ is added to the covariance function $k(\mathbf{x}, \mathbf{x}')$.

4.2 Regression

Gaussian process models fit well to the Bayesian framework, as measured data can be conveniently used to conduct inference and predict the values of new observations. Consider a simple case of noise-free training observations $\mathcal{D} = (\mathbf{X}, \mathbf{f}) = \{(\mathbf{x}^{(i)}, f^{(i)})\}_{i=1}^n$ and a task to predict the output values $\mathbf{f}^*$ at some test points $\mathbf{X}^*$. Specifying the mean as zero, the joint distribution of the training outputs $\mathbf{f}$ and test outputs $\mathbf{f}^*$ is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right). \tag{24}$$

Here, the block matrix of K's represents the matrix of covariances between all pairs of trainings points $\mathbf{X}$ and test points $\mathbf{X}^*$. The posterior distribution over functions is obtained by conditioning the joint Gaussian distribution on the observations $\mathbf{f}$, resulting in

$$p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{f}) = \mathcal{N}(\mathbf{f}^*|\mathbf{K}^*\mathbf{K}^{-1}\mathbf{f}, \mathbf{K}^{**} - \mathbf{K}^*\mathbf{K}^{-1}\mathbf{K}^{*\mathsf{T}}), \tag{25}$$

where the notation is simplified as $\mathbf{K} = K(\mathbf{X}, \mathbf{X})$, $\mathbf{K}^* = K(\mathbf{X}^*, \mathbf{X})$, $\mathbf{K}^{**} = K(\mathbf{X}^*, \mathbf{X}^*)$.

In typical modelling situations, the task is to infer function values $f(\mathbf{x})$ from observations that include noise, $y^{(i)} = f(\mathbf{x}^{(i)}) + \varepsilon^{(i)}$. By assuming the noise as additive, independent and identically distributed Gaussian noise with variance σ_n^2, and assuming a zero mean Gaussian process prior on the latent function values, the

distribution of the observed target values $\mathbf{y}$ becomes

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma_n^2 \mathbf{I}). \tag{26}$$

This leads to the joint distribution of the observed target values $\mathbf{y}$ and the function values at test points $\mathbf{f}^*$ as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 \mathbf{I} & \mathbf{K}^{*\mathsf{T}} \\ \mathbf{K}^* & \mathbf{K}^{**} \end{bmatrix}\right). \tag{27}$$

The conditional distribution of the latent values at the test points given observed data $\mathbf{y}$ is then

$$\begin{aligned} p(\mathbf{f}^* \,|\, \mathbf{X}^*, \mathbf{y}) &= \mathcal{N}(\mathbf{f}^* \,|\, \mathbb{E}[\mathbf{f}^*], \mathrm{Cov}[\mathbf{f}^*]), \text{ where} \\ \mathbb{E}[\mathbf{f}^*] &= \mathbf{K}^*[\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y}, \\ \mathrm{Cov}[\mathbf{f}^*] &= \mathbf{K}^{**} - \mathbf{K}^*[\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{K}^{*\mathsf{T}}. \end{aligned} \tag{28}$$

The predictive distribution of noisy target values $\mathbf{y}^*$ at test points is given similarly with the addition of a noise variance term $\sigma_n^2 \mathbf{I}$:

$$p(\mathbf{y}^* \,|\, \mathbf{X}^*, \mathbf{y}) = \mathcal{N}(\mathbf{y}^* \,|\, \mathbb{E}[\mathbf{f}^*], \mathrm{Cov}[\mathbf{f}^*] + \sigma_n^2 \mathbf{I}). \tag{29}$$

Equations (28) and (29) define the key predictive equations in Gaussian process regression.

4.3 Covariance function

The covariance function is an essential feature of a Gaussian process, because it determines the properties of the functions generated by the process (Rasmussen and Williams, 2006). The parameters of the covariance function are typically called the hyperparameters of the GP model. Often, also the parameters of the likelihood, such as the noise magnitude σ_n, are thought to be part of the hyperparameters. For Gaussian processes, the actual parameters can be considered to be the underlying latent values $\mathbf{f}$ at the training points. Therefore, the number of parameters increases with data.

A general term for a function that maps two inputs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ into $\mathbb{R}$ is a *kernel*. An arbitrary kernel function $k(\mathbf{x}, \mathbf{x}')$ is a valid covariance function for a Gaussian process only if the kernel is symmetric and positive semi-definite. Formally, it is

required that

$$\int k(\mathbf{x}, \mathbf{x}')f(\mathbf{x})f(\mathbf{x}')\mathrm{d}\mu(\mathbf{x})\mathrm{d}\mu(\mathbf{x}') \geq 0 \tag{30}$$

for all $f \in L^2(\mathcal{X}, \mu)$. Here, μ is a measure on the input space $\mathcal{X}$, and $L^2(\mathcal{X}, \mu)$ is the space of square-integrable functions. A covariance function is *stationary* if it is a function of the difference $\mathbf{x} - \mathbf{x}'$, thus being invariant to translations in the input space. A stationary covariance function is *isotropic* if it is also invariant to rotations by depending only on the Euclidean distance $||\mathbf{x} - \mathbf{x}'||$. Covariance functions can be combined to create new ones, and valid covariance functions are created as a sum, product, or convolution of existing covariance functions.

The main assumption that a covariance function conveys is how informative the function values at two input locations are to each other. For regression problems, using stationary covariance functions can often be justified, because it corresponds to the belief that inputs close together have similar output values no matter their location. For stationary covariance functions, a typical choice is to have the covariance decay with distance. The rate of decay is often described with a parameter l, which can be interpreted as a characteristic length-scale of the latent function.

One very common covariance function for regression is the squared exponential (SE) with a single length-scale

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}||\mathbf{x} - \mathbf{x}'||_2^2\right). \tag{31}$$

The function is infinitely differentiable, and both stationary and isotropic. Stein (2012) argues that infinite differentiability is an unrealistic assumption, but the squared exponential has still retained its popularity in regression tasks. Theoretically, regression with this covariance function is equal to Bayesian linear regression with infinitely many Gaussian basis functions.

As differentiation is a linear operation, the derivative of a Gaussian process is another Gaussian process (Solak et al., 2003; Riihimäki and Vehtari, 2010). The partial derivatives of the latent mean function at a test point $\mathbf{x}^*$ are therefore tractable, and they are equal to the mean of the derivative Gaussian process at that point. For the partial derivative with respect to the covariate x_d, the following equality holds

$$\mathbb{E}\left[\frac{\partial f^*}{\partial x_d^*}\right] = \frac{\partial \mathbb{E}[f^*]}{\partial x_d^*}. \tag{32}$$

Having observed some noisy values $\mathbf{y}$ of the original GP, the mean and variance of

the partial derivative of the latent function with respect to the covariate x_d are

$$\begin{aligned} \mathbb{E}\left[\frac{\partial f^*}{\partial x_d^*}\right] &= \frac{\partial \mathbf{K}^*}{\partial x_d^*}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \\ \mathrm{Var}\left[\frac{\partial f^*}{\partial x_d^*}\right] &= \frac{\partial^2 \mathbf{K}^{**}}{\partial x_d^* \partial x_d^*} - \frac{\partial \mathbf{K}^*}{\partial x_d^*}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\frac{\partial \mathbf{K}^{*\mathsf{T}}}{\partial x_d^*}. \end{aligned} \tag{33}$$

4.4 Model training

The previous sections have described inference with Gaussian processes for fixed values of the free parameters. However, typically these are not known, and one should also perform inference over them. To this end, one has to assign a prior for the hyperparameters and compute the joint posterior distribution of the hyperparameters $\boldsymbol{\theta}$ and the latent function values $\mathbf{f}$:

$$p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{f}, \boldsymbol{\theta})p(\mathbf{f}|\mathbf{X}, \boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{34}$$

Moreover, the predictions for future observations should be integrated over (34):

$$p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{f}, \boldsymbol{\theta})p(\mathbf{f}, \boldsymbol{\theta}|\mathbf{y})\mathrm{d}\mathbf{f}\mathrm{d}\boldsymbol{\theta}. \tag{35}$$

However, full inference is analytically intractable. A computationally convenient and common procedure is to fix the hyperparameters to point estimates $\hat{\boldsymbol{\theta}}$ given by maximizing the marginal likelihood, a method that is described in Section 3.3. The optimization of the hyperparameters is often referred to as training or fitting of the Gaussian process.

With the Gaussian likelihood, the logarithm of the marginal likelihood for a GP model is

$$\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\mathsf{T}(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K} + \sigma_n^2\mathbf{I}| - \frac{n}{2}\log(2\pi). \tag{36}$$

The three terms of the log marginal likelihood are interpretable as follows. The first term is the data-fit term and it becomes larger when the GP fits the data well. The second term is a complexity term that carries out the Occam's razor principle, and the third term is a normalization constant. Even though the log marginal likelihood is known analytically, its maximization is a non-convex optimization problem. The complexity of computing the marginal likelihood is dominated by the need to invert the $n \times n$ matrix, which has a time complexity of $\mathcal{O}(n^3)$. Once the inverse is computed, the partial derivatives with respect to the hyperparameters are only $\mathcal{O}(n^2)$ in complexity each, and a gradient based optimization is therefore computationally

efficient.

As mentioned in Section 3.3, optimizing the hyperparameters may lead to severe over-fitting to the training data. The risks of over-fitting when determining the parameters of a parametric model are well known and broadly documented. However, the risk of over-fitting when optimizing hyperparameters are discussed less often. For Gaussian process models, it has been shown that in both classification (Cawley and Talbot, 2010) and regression (Mohammed and Cawley, 2017), using the evidence framework with the squared exponential kernel results in better generalization performance when the length-scales of each covariate are equal. This is because this form has less hyperparameters and thus has less possibilities to over-fit to the training data.

5 Covariate selection

It is often difficult in practice to know which variables are useful for predicting a target variable. The task of selecting the input variables to include in the model is called covariate selection. This task may be defined in different ways, but the aim is roughly to select a subset of the covariates $\mathbf{x} = \{x_1, \ldots, x_p\}$ and maximize the predictive performance of the model. Here, it is assumed that the model is otherwise identical, i.e. the model structure is fixed and only the set of included covariates is varied. The tools of model selection from Section 3 could in principle be used also for covariate selection. However, the total amount of different variable combinations is an exponential function of the number of available input variables, which often calls for a different approach to the problem.

The covariate selection task can be understood through the definitions of covariate relevance and covariate redundance (John et al., 1994; Koller and Sahami, 1996; Guyon and Elisseeff, 2003; Yu and Liu, 2004). Strong relevance of a covariate means that removing it from the model always reduces the predictive ability. Weak relevance, on the other hand, indicates that predictive performance is sometimes reduced and sometimes not, depending on the other included covariates. A covariate that does not contain any useful information for predicting the target variable is neither strongly or nor weakly relevant. Such covariates are called irrelevant. For example, an irrelevant covariate can contain only noise that has zero impact on the target variable. Covariate redundance is closely connected to correlation between the covariates. Simply put, redundant covariates contain only information that is included also in other covariates. For example, two completely correlated covariates are redundant to each other, and only one of them is therefore useful.

Yu and Liu (2004) showed that the full set of covariates can be divided into four disjoint sets: irrelevant covariates (I), weakly relevant and redundant covariates (II), weakly relevant and nonredundant covariates (III), and strongly relevant covariates (IV). While the division is disjoint, it is not unique as the redundance of a covariate depends on which other covariates are included. They define the union of (III) and (IV) as an optimal subset, as choosing it retains an equal predictive performance compared to the full model.

Note that the above definition of the optimal subset requires that the predictive performance of the submodel is equal to the full model. However, one may be willing to select an even smaller subset if it simplifies the model, but does not drastically weaken the predictive performance. In this case, there is typically a trade-off between the predictive ability of the model and the number of covariates included. The desired compromise between simplicity of the model and predictive performance depends on

the application.

Covariate selection has several advantages. By identifying the important variables, it improves the interpretability of the model and the whole problem. Additionally, it reduces the cost of collecting and storing data as well as making measurements in the future, as the removed variables need not be measured. However, one should be careful with covariate selection, because the large number of possible variable combinations increases the risk of over-fitting to the available data during the selection process. In other words, it is very probable that some covariate combinations fit especially well to the particular data set even though they do not generalize well to new data. This phenomenon is called *selection induced bias* (Reunanen, 2003; Cawley and Talbot, 2010; Vehtari and Ojanen, 2012; Piironen and Vehtari, 2017).

Covariate selection is closely connected to covariate extraction, as both are different approaches to dimensionality reduction, i.e. reducing the number of random variables in a model. Covariate extraction uses linear or nonlinear transformations to the data to reduce the dimensionality, while constructing a set of relevant features. One famous feature extraction method is principal component analysis, which uses orthogonal transformations to produce linearly uncorrelated covariates. Feature selection and feature extraction both have their own merits, but because covariate selection preserves the original covariates, it is useful in applications requiring interpretation of the model. For a thorough introduction to feature extraction, see (Guyon et al., 2008).

5.1 Selection methods

A myriad of different methods have been developed specifically for covariate selection, and it is not feasible to introduce them all. The different approaches can be roughly categorized into subset evaluation methods and covariate ranking methods. Subset evaluation methods directly assess the performance of subsets, and can therefore remove also redundant covariates. The downside of subset evaluation methods is a much higher computational complexity compared to covariate ranking methods. Even with greedy sequential search methods, the complexity is still $\mathcal{O}(p^2)$ for p covariates.

Covariate ranking methods estimate the relevance of each covariate individually, and assign to each a weight corresponding to their relevance. A subset of suitable size can then be chosen based on the ranking. This approach has a linear time complexity with respect to the number of covariates, and is therefore useful for high-dimensional data. The drawback of individual evaluation methods is that they are often ineffective at removing redundant covariates, because they are likely ranked

approximately equally good.

One approach to covariate ranking is to assess the expected change in the target variable as the input variables are changed. For example, Stolzenberg (1980) and Härdle and Stoker (1989) have examined the expected change by estimating the average partial derivatives of a regression curve. In a simple linear regression model, the partial derivatives are analytically tractable and are given by the regression coefficients. However, in more complicated models, the derivatives need to be estimated. Gelman and Pardoe (2007) extended the idea of the expected difference from the partial derivatives to a more general approach. They define a predictive comparison of a covariate x_j as the difference quotient of the expected value of the model predictions:

$$\delta_j(x_j^{(2)} \to x_j^{(1)}, \mathbf{x}_{-j}, \mathbf{w}) = \frac{\mathbb{E}(y|x_j^{(2)}, \mathbf{x}_{-j}, \mathbf{w}) - \mathbb{E}(y|x_j^{(1)}, \mathbf{x}_{-j}, \mathbf{w})}{x_j^{(2)} - x_j^{(1)}}. \tag{37}$$

Here, $\mathbf{x}_{-j}$ is the value of all input variables except x_j, and $\mathbf{w}$ are the parameters of the model. The predictive comparison is not equal to the partial derivative with respect to x_j, because the limit $x_j^{(2)} - x_j^{(1)} \to 0$ is not taken. The average relevance of a covariate x_j can be computed by averaging the predictive comparison over $\mathbf{w}$, $\mathbf{x}_{-j}$, $x_j^{(2)}$, and $x_j^{(1)}$, where only increasing transitions $x_j^{(1)} < x_j^{(2)}$ are considered. The practical disadvantage of the average predictive comparison is the expensive summation over all the variables.

5.2 Covariate selection with Gaussian process models

An alternative form of the squared exponential covariance function (31) is one with separate length-scale parameters for each input variable

$$k_{\text{ARD}-\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{p}\frac{(x_i - x_i')^2}{l_i^2}\right). \tag{38}$$

While the common parameter σ_f determines the overall variability, the different length-scale parameters l_i for each of the p input dimensions allow the generated functions to vary at different scales along the covariates. This form is sometimes more favourable compared to (31) because it is a less informative prior over the functions and is thus more flexible. The separate length-scales can be utilized for covariate selection, and this section presents different methods that have been used in the literature.

5.2.1 Sparsity promoting priors

As Gaussian processes are inherently Bayesian models, covariate selection is often carried out using so-called *Bayesian variable selection* methods. In the Bayesian framework, covariate selection can be done with priors that promote sparsity, i.e. include the assumption that not all of the covariates are included. One such prior is the spike-and-slab prior. For a linear regression model $\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon}$ for example, the spike-and-slab prior on a single weight parameter w_j has the form

$$p(w_j|\gamma_j) = \gamma_j \tilde{p}(w_j) + (1 - \gamma_j)\delta_0(w_j), \tag{39}$$

where δ_0 is the Dirac delta function at zero and γ_j is the parameter that controls the exclusion probability of the weight (Mitchell and Beauchamp, 1988). Thus, with probability $(1 - \gamma_j)$ the covariate x_j is excluded from the model by setting the weight to zero, and with probability γ_j it is included and has a "slab" prior $\tilde{p}(w_j)$. The slab part can be a noninformative Gaussian prior, for example. The spike-and-slab is a common prior for achieving sparsity and is sometimes referred to as the golden standard from a Bayesian perspective (George and McCulloch, 1993; Titsias and Lázaro-Gredilla, 2011).

In the Gaussian process framework, covariate selection can be done by assigning spike-and-slab type priors on the length-scale parameters of the covariance function. To achieve this, one has to parametrize the covariance function differently to obtain a nonzero probability for the covariance of a particular dimension to be exactly zero. Savitsky et al. (2011) use a squared exponential (38) kernel and a spike-and-slab type prior of the form

$$p(\rho_k|\gamma_k) = \gamma_k \mathrm{U}(\rho_k|0, 1) + (1 - \gamma_k)\delta_1(\rho_k), \tag{40}$$

where γ_k is the parameter controlling the probability for the covariate to be irrelevant, and ρ_k is a parameter that controls the covariance of the latent values corresponding to covariate x_j, which is parametrized in such a way that its domain is $[0, 1]$. In this parametrization, the exponent of (38) has terms $-\log(\rho_i)$ instead of l_i^{-2}. Spike-and-slab type sparsifying priors have been used for covariate selection for Gaussian process models by Vehtari (2001), Linkletter et al. (2006), and Savitsky et al. (2011).

5.2.2 Automatic relevance determination

The separate length-scales of (38) represent the nonlinearity of each covariate. However, this nonlinearity value can also be used as an estimate for predictive relevance,

which is known as automatic relevance determination (ARD). After optimizing the hyperparameters, ARD allows the relevance of a covariate to be inferred from the inverse of its corresponding length-scale parameter. This interpretation is based on the remark that taking the length-scale to infinity renders the generated functions flat. The ARD method was originally proposed for estimating the relevance of inputs in multilayer perceptrons by MacKay (1994) and Neal (1995).

The use of length-scales as a measure of relevance has two problems. Firstly, the length-scale parameters are not well identified (Zhang, 2004). This increases variance of the relevance estimate given by ARD. Especially when a length-scale is large compared to the scale of the data, the generated function is essentially linear with respect to the corresponding covariate. Thus, increasing the length-scale further does not significantly alter the likelihood. Secondly, ARD severely overestimates the relevance of nonlinear covariates over linear or near-linear covariates of equal relevance in the squared error sense, which has been demonstrated for multilayer perceptrons (Lampinen and Vehtari, 2001) as well as Gaussian processes (Piironen and Vehtari, 2016).

5.2.3 Projection predictive covariate selection

Piironen and Vehtari (2016) devised a method for projecting the information of a full GP model onto simpler submodels with less covariates by revising an idea originally introduced for generalized linear models (Goutis and Robert, 1998; Dupuis and Robert, 2003). The projection is made by optimizing the hyperparameters of the submodel so that the posterior latent Kullback-Leibler divergence at the training points is minimized. Essentially, the full GP model is treated as a reference model, and the submodel is chosen so that predictions change minimally. The projection method requires some search heuristic during the projection, and each step requires fitting multiple GPs during the search. It is thus computationally quite expensive, but it has been shown to outperform automatic relevance determination by choosing better submodels in simulated and real world regression problems (Piironen and Vehtari, 2016). In generalized linear models, the projection approach was shown to be superior to many other variable selection methods (Piironen and Vehtari, 2017).

6 Methods

The focus of this thesis is to study covariate selection methods for Gaussian process models. This section will introduce and motivate two novel methods, which are the main contribution of this thesis. The methods are introduced as possible alternatives to automatic relevance determination, introduced in Section 5.2.2, which is the prevalent covariate selection method for GPs.

The methods are named KL and VAR, arising from the utilization of KL divergence as well as the estimation of the variance of the GP mean. The motivation for these methods is to avoid the drawbacks of ARD by taking a different approach than examining the length-scale parameter of the covariance function. The methods exploit the fact that the posterior predictive distribution of a GP model is analytically tractable. By examining the distribution at the training data points, the average predictive relevance can be effectively estimated without increasing the computational complexity prohibitively over ARD.

6.1 KL divergence as a relevance measure

The Kullback-Leibler divergence (15) is prevalent in a multitude of contexts in statistics and machine learning. It is a measure of how one probability distribution diverges from another probability distribution, and is thus an asymmetric measure. In the KL method, Kullback-Leibler divergence is utilized to measure the divergence of the posterior predictive distribution of a Gaussian process model at an input point when it is perturbed. Via sensitivity analysis of the posterior predictive distribution, the predictive relevance of covariates can be estimated.

In GP models with a Gaussian likelihood assumption, the posterior predictive distribution at a single test point is a univariate normal distribution. The predictive equation for the latent values is given in (28). Denoting the mean and variance of the predictive distribution as $\mathbb{E}[f(\mathbf{x})] = \mu(\mathbf{x})$ and $\text{Var}[f(\mathbf{x})] = \sigma^2(\mathbf{x})$, the KL divergence (15) from the predictive distribution at $\mathbf{x}$ to the distribution at $\mathbf{x}'$ is (Kullback and Leibler, 1951)

$$\begin{aligned} &D_{\text{KL}}(\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \,||\, \mathcal{N}(\mu(\mathbf{x}'), \sigma^2(\mathbf{x}'))) \\ &= \log \frac{\sigma(\mathbf{x}')}{\sigma(\mathbf{x})} + \frac{\sigma^2(\mathbf{x}) + (\mu(\mathbf{x}) - \mu(\mathbf{x}'))^2}{2\sigma^2(\mathbf{x}')} - \frac{1}{2}. \end{aligned} \tag{41}$$

As this is a quadratic function of the difference in their means, taking the square root of the KL divergence draws an analogy to the derivative of the GP mean. By considering the total-variation distance of Pinsker's inequality, the final measure for

the difference of predictive distributions is chosen as (Simpson et al., 2017)

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{2\, D_{\mathrm{KL}}(\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \,||\, \mathcal{N}(\mu(\mathbf{x}'), \sigma^2(\mathbf{x}')))}. \tag{42}$$

Because KL divergence is always nonnegative, the measure is also a nonnegative real number.

Consider a Gaussian process with a squared exponential kernel and Gaussian observation model fitted to some training data $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ with n observations and p covariates. As the squared exponential is infinitely smooth, the partial derivative of the mean of the GP with respect to a single covariate x_j at an arbitrary point $\mathbf{a}$ can be obtained as the limit of the difference quotient

$$\begin{aligned}
&\lim_{\Delta \to 0} \frac{\mathbb{E}[f(a_1, \ldots, a_{j-1}, a_j + \Delta, a_{j+1}, \ldots, a_p)] - \mathbb{E}[f(a_1, \ldots, a_j, \ldots, a_p)]}{\Delta} \\
&= \lim_{\Delta \to 0} \frac{\mathbb{E}[f(\mathbf{a} + \Delta_j)] - \mathbb{E}[f(\mathbf{a})]}{\Delta} = \lim_{\Delta \to 0} \frac{\mu(\mathbf{a} + \Delta_j) - \mu(\mathbf{a})}{\Delta} \\
&= \frac{\partial}{\partial x_j} \mathbb{E}[f(\mathbf{a})].
\end{aligned} \tag{43}$$

Keeping the analogy with the derivative and taking the limit of the difference quotient of the difference measure (42) yields

$$\begin{aligned}
&\lim_{\Delta \to 0} \frac{d(\mathbf{a}, \mathbf{a} + \Delta_j)}{\Delta} \\
&= \lim_{\Delta \to 0} \frac{\sqrt{2 D_{\mathrm{KL}}(\mathcal{N}(\mu(\mathbf{a}), \sigma^2(\mathbf{a})) \,||\, \mathcal{N}(\mu(\mathbf{a} + \Delta_j), \sigma^2(\mathbf{a} + \Delta_j)))}}{\Delta} \\
&= \lim_{\Delta \to 0} \sqrt{\frac{2(\log \frac{\sigma(\mathbf{a}+\Delta_j)}{\sigma(\mathbf{a})} + \frac{\sigma^2(\mathbf{a}) + (\mu(\mathbf{a}) - \mu(\mathbf{a}+\Delta_j))^2}{2\sigma^2(\mathbf{a}+\Delta_j)} - \frac{1}{2})}{\Delta^2}}.
\end{aligned} \tag{44}$$

Because the square root is continuous over its whole domain, the limit can be moved inside and the above equals

$$\begin{aligned}
&\sqrt{\lim_{\Delta \to 0} \frac{2(\log \frac{\sigma(\mathbf{a}+\Delta_j)}{\sigma(\mathbf{a})} + \frac{\sigma^2(\mathbf{a}) + (\mu(\mathbf{a}) - \mu(\mathbf{a}+\Delta_j))^2}{2\sigma^2(\mathbf{a}+\Delta_j)} - \frac{1}{2})}{\Delta^2}} \\
&= \sqrt{0 + \lim_{\Delta \to 0} \frac{(\mu(\mathbf{a}) - \mu(\mathbf{a} + \Delta_j))^2}{\Delta^2 \sigma^2(\mathbf{a} + \Delta_j)}} \\
&= \lim_{\Delta \to 0} \frac{\mu(\mathbf{a} + \Delta_j) - \mu(\mathbf{a})}{\Delta \sigma(\mathbf{a} + \Delta_j)} \\
&= \frac{1}{\sigma(\mathbf{a})} \frac{\partial}{\partial x_j} \mathbb{E}[f(\mathbf{a})].
\end{aligned} \tag{45}$$

The limit exists, because with noisy observations $\mathbf{y}$, the predictive standard deviation $\sigma(\mathbf{a})$ is never zero. Thus, perturbing an input point $\mathbf{a}$ with respect to one covariate by Δ and taking the limit $\Delta \to 0$ yields a neat form to (42) as the partial derivative of the GP mean divided by the standard deviation of the predictive distribution. However, the formulation (42) of the relevance measure is more general and does not require the observation model to be Gaussian.

Because the average derivative is correlated with predictive relevance, this will be used for estimating the predictive relevance of covariates. The additional benefit of the above formulation is that the partial derivatives are weighted by the uncertainty of the predictive distribution at each point. Giving less weight to uncertain predictions improves the accuracy of the estimation. Taking the square root of the KL divergence is also beneficial from a practical point of view. When the measure (42) is estimated at multiple points, the square root is less sensitive to variations in the standard deviation between the points. Thus it estimates the average relevance better than without the square root.

The above methodology can be used to estimate the predictive relevance of covariates at any point of the input space. Due to the curse of dimensionality, it is not trivial how one should choose the points if one wants to estimate the predictive relevance of covariates. In this thesis, the estimates are computed at each training point by perturbing the training point separately in each input dimension. This will produce a good representation of the average relevance with relatively little computational burden. By always computing the KL divergence from the training point to a nearby point, the method naturally complies with the asymmetry of the KL divergence. The partial derivatives of the GP mean can be computed analytically, as discussed in Section 4.3. However, in this thesis we fix the perturbation distance Δ to some small number, and compute the relevance estimates directly via the KL divergence. Because of this, the relevance measure (42) is divided by Δ:

$$\tilde{d}(\mathbf{x}, \mathbf{x}') = \frac{\sqrt{2\, D_{\mathrm{KL}}(\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \,||\, \mathcal{N}(\mu(\mathbf{x}'), \sigma^2(\mathbf{x}')))}}{\Delta}. \tag{46}$$

6.1.1 Choice of perturbation distance

The choice of the perturbation distance Δ has to be determined based on the distribution of input data. When Δ is chosen small enough, the KL method is not sensitive to the size of the perturbation. The results of this thesis are computed with Δ approximately 0.0001 times the standard deviation of the inputs, and varying the distance for two orders of magnitude above and below this value did not alter the results. However, very small values should be avoided because of potential numerical

errors.

6.2 Variance of the predictive mean

This section presents the second new method for ordering covariates based on their predictive relevance. This method is denoted as VAR because it is based on estimating the variance of each component of the latent function of a fitted Gaussian process. In order to efficiently achieve this, the distribution of the input variables will be estimated. By assuming that the inputs have a joint Gaussian distribution $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the conditional distribution of one covariate, given the value of the others, is a univariate normal distribution. If the joint distribution of the input variables, with covariate x_j separated, is denoted as

$$\begin{bmatrix} \mathbf{x}_{-j} \\ x_j \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{-j} \\ \mu_j \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{-j,-j} & \boldsymbol{\sigma}_{j,-j} \\ \boldsymbol{\sigma}_{-j,j} & \sigma_{j,j} \end{bmatrix} \right), \tag{47}$$

the conditional distribution of the covariate x_j is given by

$$\begin{aligned} x_j \,|\, \mathbf{x}_{-j} &\sim \mathcal{N}(\mu_a, \sigma_a^2), \\ \mu_a &= \mu_j + \boldsymbol{\sigma}_{j,-j} \boldsymbol{\Sigma}_{-j,-j}^{-1} (\mathbf{x}_{-j} - \boldsymbol{\mu}_{-j}), \\ \sigma_a^2 &= \sigma_{j,j} - \boldsymbol{\sigma}_{j,-j} \boldsymbol{\Sigma}_{-j,-j}^{-1} \boldsymbol{\sigma}_{-j,j}. \end{aligned} \tag{48}$$

The subscript j refers to selecting the row or column j from $\boldsymbol{\mu}$ or $\boldsymbol{\Sigma}$, whereas the subscript $-j$ refers to excluding them. Using equation (48), the relevance of covariate x_j at an arbitrary point can be estimated by computing the variance of the predictive mean along the covariate with Gauss-Hermite quadrature (Golub and Welsch, 1969). A good estimate for the overall relevance of each covariate is achieved by repeating this procedure at each training point.

The predictive distribution of a single-output GP in one test point is a univariate Gaussian. Let us denote the mean of this distribution as $\mu_*(\mathbf{x})$. Variance of the mean along a covariate x_j is then given by integrating over the conditional Gaussian of equation (48)

$$\begin{aligned} \mathrm{Var}[\mu_{*,j}(x_j|\mathbf{x}_{-j})] = &\int \mu_{*,j}^2(x_j)\, \mathcal{N}(x_j \,|\, \mu_a, \sigma_a^2)\, \mathrm{d}x_j \\ &- \left(\int \mu_{*,j}(x_j)\, \mathcal{N}(x_j \,|\, \mu_a, \sigma_a^2)\, \mathrm{d}x_j \right)^2. \end{aligned} \tag{49}$$

With a change of variables $k = (x_j - \mu_a)/(\sqrt{2}\sigma_a)$, the variance takes the form

$$\begin{aligned}\mathrm{Var}[\mu_{*,j}(x_j|\mathbf{x}_{-j})] &= \int \mu_{*,j}^2(\sqrt{2}\sigma_a k + \mu_a)\frac{e^{-k^2}}{\sqrt{\pi}}\,\mathrm{d}k \\ &\quad - \left(\int \mu_{*,j}(\sqrt{2}\sigma_a k + \mu_a)\frac{e^{-k^2}}{\sqrt{\pi}}\,\mathrm{d}k\right)^2 \\ &\approx \pi^{-1/2}\sum_{i=1}^{N} w_i\,\mu_{*,j}^2(\sqrt{2}\sigma_a k_i + \mu_a) \\ &\quad - \pi^{-1}\left(\sum_{i=1}^{N} w_i\,\mu_{*,j}(\sqrt{2}\sigma_a k_i + \mu_a)\right)^2.\end{aligned} \tag{50}$$

where w_i and k_i are the weights and evaluation points of the Gauss-Hermite quadrature, and N is the number of these points.

6.2.1 Precision matrix estimation

In order to compute the conditional distribution of one covariate (48), the full sample mean $\boldsymbol{\mu}$ and sample covariance matrix $\boldsymbol{\Sigma}$ of the inputs must be estimated from the training inputs. In addition, a submatrix of the covariance matrix, with one row and column removed, must be inverted. The details for computing the inverse of the submatrix efficiently from the full precision matrix are explained in Appendix A. This sections discusses methods for estimating the covariance matrix $\boldsymbol{\Sigma}$ and the precision matrix $\mathbf{P} = \boldsymbol{\Sigma}^{-1}$ from n samples $\{\mathbf{x}^{(i)}\}_{i=1}^n$ of a p-dimensional random variable $\mathbf{x}$.

The population covariance matrix $\boldsymbol{\Sigma}$ of the random variable $\mathbf{x}$ is defined as (Johnson and Wichern, 2007)

$$\boldsymbol{\Sigma} = \mathrm{Cov}[\mathbf{x}, \mathbf{x}] = \mathrm{Var}[\mathbf{x}] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\mathsf{T}]. \tag{51}$$

This can be estimated with the sample covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-1}\mathbf{S}, \tag{52}$$

where $\mathbf{S}$ is the scatter matrix, defined as

$$\mathbf{S} = \sum_{i=1}^{n}(\mathbf{x}^{(i)} - \overline{\mathbf{x}})(\mathbf{x}^{(i)} - \overline{\mathbf{x}})^\mathsf{T}, \tag{53}$$

and $\overline{\mathbf{x}}$ is the sample mean

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}. \tag{54}$$

Both $\overline{\mathbf{x}}$ and $\hat{\mathbf{\Sigma}}$ are unbiased estimators for the mean and covariance for any distribution of $\mathbf{x}$, that is $\mathbb{E}[\overline{\mathbf{x}}] = \mathbb{E}[\mathbf{x}]$ and $\mathbb{E}[\hat{\mathbf{\Sigma}}] = \mathbf{\Sigma}$. For specific distributions, different estimators can be defined with varying properties. If $\mathbf{x}$ follows a normal distribution, the maximum likelihood estimator for the covariance is

$$\hat{\mathbf{\Sigma}}_{\mathrm{N-ML}} = \frac{1}{n}\mathbf{S}. \tag{55}$$

The most basic way to estimate $\mathbf{P} = \mathbf{\Sigma}^{-1}$ is to directly invert an estimator of $\mathbf{\Sigma}$, such as the sample covariance matrix. However, the sample covariance matrix does not accurately estimate the true eigenvalues of $\mathbf{\Sigma}$. This typically manifests as overestimation of large eigenvalues and underestimation of small eigenvalues, and causes large errors during the inversion. There are several ways to account for these errors. Two commonly used approaches are sparsifying and shrinkage. Sparsifying methods make the assumption that the precision matrix is sparse, and aim to achieve this somehow. For example, the popular graphical lasso method imposes an L_1 penalty for the estimation of the precision matrix to achieve sparsity (Friedman et al., 2008). Shrinkage estimators, on the other hand, aim to shrink the eigenvalues of the sample covariance matrix by constructing a linear combination of an estimator matrix and some target matrix, thus improving the stability of the inversion. Very often, this target matrix is the identity matrix.

In this thesis, only data sets with more data points than input dimensions are considered. The precision matrix is estimated by inverting the maximum likelihood estimator (55), where a small identity matrix is added to increase the numerical stability. In the absence of linearly dependent components in the inputs, the ML estimator is positive definite and its inverse can be computed using the Cholesky decomposition.

The difficulties arising from estimating and inverting the covariance of the inputs result in inaccuracies to the relevance estimation via the VAR method. However, the experiments in Section 7 show that the method works well in practice even when the input distribution is far from a normal distribution. One way to avoid the errors caused by inversion altogether is to integrate the equation (49) over the marginal distribution of x_j instead of the conditional distribution. However, this will create another source of error because the distribution is not correct.

6.3 Computational complexity

The exact inference with Gaussian processes has complexity $\mathcal{O}(n^3)$ for a data set with n observations. This hinders their applicability especially in large data sets. Once a

full GP model is fitted, ordering covariates using ARD comes about automatically, requiring no additional computations. By a projection approach (Piironen and Vehtari, 2016), the covariates can be ordered more effectively, but the drawback is an increase in complexity to $\mathcal{O}(p^2n^3)$, where p is the dimension of the inputs.

The complexity of Gaussian process inference arises from the unavoidable matrix inversion. However, the same inverse can be used for making an arbitrary number of predictions at new test points, achieved by solving triangular systems, which are only $\mathcal{O}(n^2)$ in complexity. The KL method needs to make one prediction at each training point, which is compared to two predictions for every input dimension that are a distance Δ above and below the training point. Thus, it requires computing $2p + 1$ predictions at every training point, giving it a total complexity of $\mathcal{O}(p{\cdot}n{\cdot}n^2) = \mathcal{O}(pn^3)$.

The VAR method, on the other hand, requires computing as many predictions as the chosen number of quadrature points for every dimension and training point. This number can be chosen to be a small constant, thus keeping the total complexity at $\mathcal{O}(pn^3)$. In addition to this, the method requires computing the inverse of the sample covariance submatrix of the inputs, $\mathbf{\Sigma}_{-j,-j}$, for each of the p covariates. Taking advantage of the positive definiteness of the full covariance matrix, the Cholesky decomposition of it, $\mathcal{O}(p^3)$ in complexity, needs to be computed only once per training set. Then the Cholesky decomposition for each submatrix $\mathbf{\Sigma}_{-j,-j}$ is obtained with a rank one update from the full covariance matrix, resulting in p rank one updates of complexity $\mathcal{O}(p^2)$. The details of the rank one updating are described in Appendix A. Thus, the full complexity of the variance method is $\mathcal{O}(pn^3 + p^3)$.

7 Experiments

This section presents a range of numerical experiments, where the covariate selection methods introduced in Section 6 are compared to automatic relevance determination. First, a toy dataset will be constructed that will be used to highlight one of the key drawbacks of automatic relevance determination. Second, the performance of the three covariate selection methods is evaluated with four freely available real world data sets. Finally, the differences in the results are explained via the theoretical properties of the methods. All of the Gaussian process models were constructed and trained using the GPy (2012) framework.

7.1 Toy dataset

Piironen and Vehtari (2016) evaluated the relevance determination capability of ARD by considering a toy dataset with eight covariates and a target variable constructed as a sum of eight independent components with varying degrees of nonlinearity. They showed that ARD severely favours covariates with a nonlinear response compared to linear covariates that are equally relevant in the squared error sense. In this section, a similar toy example is constructed, but besides just uniformly distributed input data, also normally distributed inputs are considered. The toy dataset is defined as follows:

$$\begin{aligned} x_j \sim \mathrm{U}(-1,1) \quad \textbf{or} \quad x_j \sim \mathcal{N}(0, 0.4^2) \;, j = 1, \ldots, 8, \\ y = f_1(x_1) + \ldots + f_8(x_8) + \varepsilon, \\ \varepsilon \sim \mathcal{N}(0, 0.3^2), \\ f_j(x_j) = A_j \sin(\phi_j x_j), \end{aligned} \tag{56}$$

where the sine coefficients ϕ_j are equally spaced between $\pi/10$ and π, and the scaling factors A_j are such that the variance of each $f_j(x_j)$ is one. The variances and scaling factors are thus different depending on the distribution of the inputs x_j. The functions $f_j(x_j)$ are presented in Figure 1 for uniformly distributed inputs (black) and normally distributed inputs (red).

For both the toy datasets, a Gaussian process model was constructed with the covariance function being the ARD-SE kernel (38) with an added constant term. 300 input points were sampled, and the output values y were computed according to equation (56). After optimizing the hyperparameters to the maximum of the marginal likelihood, the relevance of each covariate was estimated either directly using ARD, or by averaging the KL and VAR relevance estimates from each of

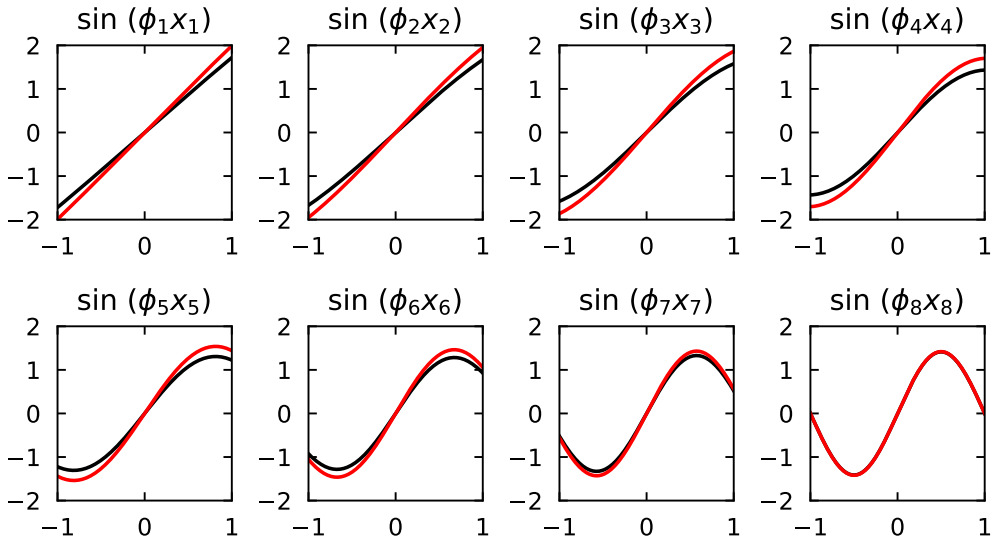

Figure 1: Latent functions $f_j(x_j), j = 1, \ldots, 8$ used to generate the two toy data sets (56). The black lines represent the dataset with uniform inputs and the red lines represent normally distributed inputs. Each latent function is scaled to unit variance according to its corresponding input distribution.

the 300 training points. The Gauss-Hermite integrations were computed using 11 quadrature points. The relevance estimates were scaled so that the largest of them was one. The averaged results of 200 random repetitions are presented in Figure 2 for the two examples with inputs distributed uniformly (top) and normally (bottom). Error bars representing 95% confidence intervals are indistinguishably small.

Figure 2 demonstrates that in the toy example with uniform inputs, all three methods prefer inputs with a nonlinear response to some extent over linear ones. However, the preference in the two alternative methods is not as severe as with ARD, which assigns relevance values close to zero for half of the covariates. The bottom figure, representing the toy example with Gaussian distributed inputs, shows that the KL and VAR methods estimate almost equal relevances for each covariate. Overall, the toy experiment shows that the KL and VAR methods are notably better than ARD in identifying the true equal relevance of the covariates with different amounts of nonlinearity.

In the toy example above, all covariates are equally relevant for prediction. In order to compare how the methods treat irrelevant covariates, the dataset was extended with 42 totally irrelevant covariates that have zero impact on the target variable. The extended toy example thus has 50 total covariates, only 8 of which are relevant, and they are similar as in (56). The average relevance values from 400 data realizations are shown in Figure 3. The results show that all methods give

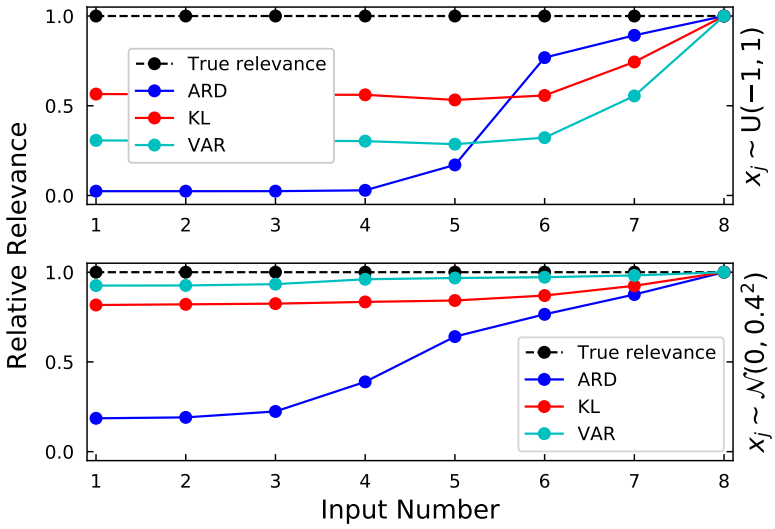

Figure 2: Relevance estimates for eight covariates in the two toy examples (56) with uniformly distributed inputs (top) and normally distributed inputs (bottom). The estimates are computed with ARD (blue), KL divergence (red), and variance of the predictive mean (cyan). The results are averaged from 200 random data sets and scaled such that the most relevant covariate has a relevance of one. Error bars representing 95% confidence intervals are indistinguishable.

negligibly small relevance values to all of the irrelevant covariates. For the relevant covariates, the relevance values are actually more equal compared to Figure 2. This is a result of the fact that some of the hyperparameter optimizations failed and resulted in a GP where some of the nonlinear covariates have a linear response. Because there are much more hyperparameters to optimize than in the small toy data set, the optimization is much more difficult. This greatly increases the variance in the relevance estimates of the most nonlinear terms. Figure 4 presents the results when some random realizations where the optimization gravely failed are removed. In this case, the results are very close to Figure 2 for the relevant covariates.

7.2 Real world data

7.2.1 Data sets

This section presents the results of experiments, where the performance of the different methods was evaluated using four data sets obtained from the UCI machine learning repository (Lichman, 2013). The data sets are summarized in Table 1. The datasets were pre-processed and normalized to obtain reasonable regression tasks without any missing data. The price was used as the target variable for the Automobile data set, and compressive strength for the Concrete Slump dataset. All data sets can be

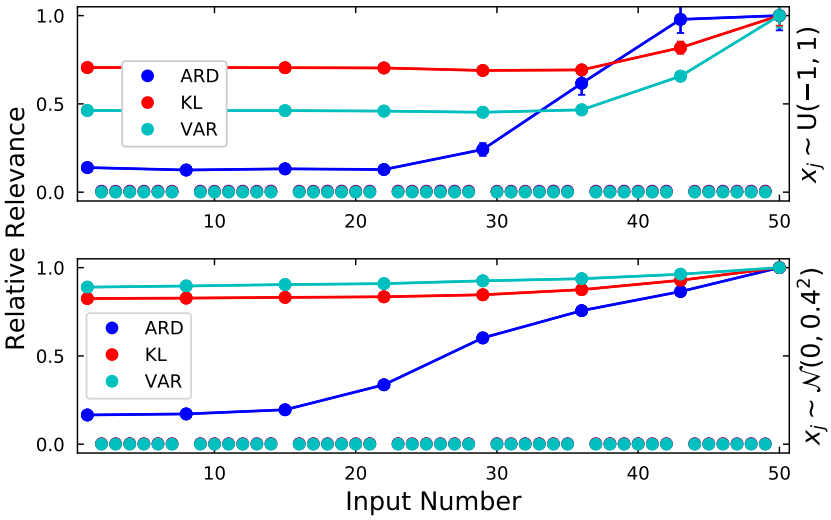

Figure 3: Relevance estimates for 50 covariates in the toy model with 8 equally relevant covariates and 42 irrelevant covariates. The estimates are computed with ARD (blue), KL divergence (red), and variance of the predictive mean (cyan). The 8 relevant covariates are joined with a line, and range from linear (input 1) to nonlinear (input 50) as in (56). The results are averaged from 400 random data sets and scaled such that the most relevant covariate has a relevance of one. The error bars represent 95% confidence intervals.

modelled quite well with a Gaussian process model with the ARD-SE covariance function, and they are thus good examples for evaluating the performance of the covariate selection methods.

Table 1: Summary of real world dataset parameters: number of covariates p, data points n_{tot}, and training points used n.

Dataset	p	n_{tot}	n	Reference
Concrete Slump Test	7	103	80	Yeh (2007)
Boston Housing	13	506	300	
Automobile	38	193	150	
Crime	102	1992	400	Redmond and Baveja (2002)

7.2.2 Predictive performance

In predictive feature selection, finding out the relevance ranking for the covariates is more important than the differences in the relevance values considered in Figure 2. To this end, we tested the predictive performance of submodels that included covariates

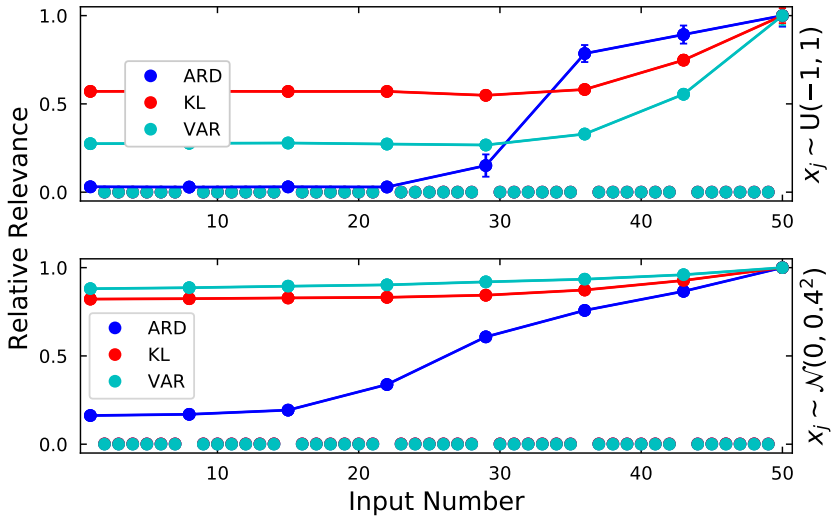

Figure 4: Relevance estimates for 50 covariates in the toy model with 8 equally relevant covariates and 42 irrelevant covariates. The results are computed from 400 data realizations, but cases where hyperparameter optimization has failed gravely, are removed. The error bars represent 95% confidence intervals.

chosen by each method. For each method, a Gaussian process model with a sum of constant and squared exponential (38) kernels as a covariance function was used. The model was first fitted with all p covariates included, and then a similar submodel was fitted with only part of the most relevant covariates included, according to the relevance ranking given by the particular method. A total of 50 repetitions was performed, each time splitting the data into random training and test sets with the number of training points shown in Table 1. Both the full model and submodels were trained on the training set, and the predictive performance of the methods was evaluated by computing the mean log-predictive densities (MLPDs) using the independent test set. The mean log-predictive density is simply the average of the log-probabilities of the posterior predictive distribution at the test points. In terms of the predictive performance estimation framework presented in Section 3.1, it is the hold-out utility (17) using the logarithmic score utility function (12).

The mean log-predictive densities of the test sets are presented in Figure 5 as a function of the number of covariates included in the submodel. A plot for each data set contains results when the covariates are sorted using ARD (blue), KL divergence (red), and variance of the predictive mean (cyan). The GP models are fitted by optimizing the hyperparameters to the maximum of the hyperparameter posterior distribution, with half-t distribution as the prior for the noise and signal magnitudes, and inverse-gamma distribution for the length-scales. The inverse-gamma was chosen because it has a sharp left tail that penalizes very small length-scales, but its long right tail

allows the length-scales to become large (Stan Development Team, 2017). The plots for the Automobile and Crime sets are shown only up to a point where the predictive performance saturates. The full model MLPD, presented as a horizontal line, was computed by sampling 100 values for every training set from the hyperparameter posterior using Hamiltonian Monte Carlo (Duane et al., 1987).

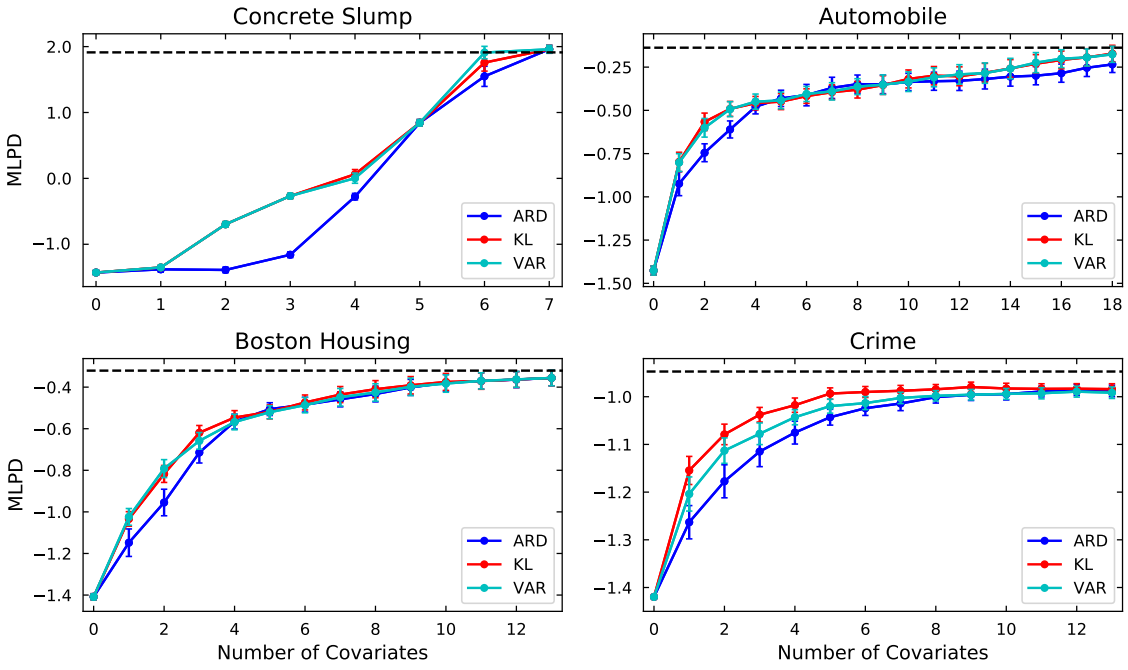

Figure 5: Mean log-predictive densities (MLPDs) computed from independent test data sets for submodels as a function of covariates included in the submodel. The error bars represent 95% confidence intervals from 50 repetitions. Blue depicts covariates sorted using ARD, red and cyan depict the KL and VAR methods, respectively. The dashed horizontal line depicts the MLPD of the full model with hyperparameters sampled using Hamiltonian Monte Carlo.

The results show that in all four data sets, The KL and VAR methods generate a slightly better ordering for the covariates than ARD does, resulting in submodels with improved predictive performance on unseen test data. The improvement is most distinct in the first three or four covariates in all the data sets. This is because ARD picks the most nonlinear covariates first by definition, while the other methods are able to identify covariates that are more relevant for prediction, albeit more linear. After the initial improvement, the ordering in the latter covariates is never worse than for ARD. Despite making the assumption of normally distributed inputs, the VAR method performs well even in the Automobile data set which has a large number of binary variables.

7.2.3 Consistency of relevance estimation

The weak identifiability of the length-scale parameter of the covariance function increases the variation in the relevance estimate given by ARD. To assess this variation, this section examines how consistently each method determines the relevance ranking of covariates between different random training sets. For every covariate, the relevance ordinal numbers given by each of the 50 training sets were computed from the four real world data sets. The plots of the ordinal numbers given by ARD, KL, and VAR methods are presented in Figure 6. The ordinal numbers are plotted for seven covariates, which were picked such that they were the most relevant on average according to the VAR method. The markers in each ordinal number slot are jittered horizontally to visualize the number of points in each slot.

Figure 6 shows that the KL and VAR produce the relevance sequence more consistently compared to ARD. By comparing this plot with Figure 5, it becomes evident that the reason for the improved predictive performance is partly a result of improved consistency. For example, the better performance in the Concrete Slump data for the submodel with 6 covariates is strictly the result of choosing covariate 5 more often than covariate 6. The Housing data plot shows that while both KL and VAR methods pick covariate 5 as the most relevant in a majority of training samples, ARD is less consistent, choosing covariates 4, 7, and 12 almost equiprobably. After the first choice, there is variance in every method, but it is largest with automatic relevance determination.

The correlation between consistency in Figure 6 and predictive performance in Figure 5 highlights one of the two major problems with ARD. Because the length-scale parameters are weakly identified, the relevance estimates have a lot of variance. This results in less consistent relevance ordering which deteriorates the predictive performance of the chosen submodels.

Figure 6 shows how the improvement in the predictive performance of submodels with 1-4 covariates is partly a result of more consistent covariate ranking. However, it does not show how consistent each method is after the first 7 choices. In order to assess the consistency of all the choices, the entropy of each subsequent covariate choice was computed from 50 training sets. For example, the entropy of the first covariate choice is

$$H_1(x) = -\sum_{i=1}^{50} p(x_i) \log p(x_i), \tag{57}$$

where $p(x_i)$ is the proportion of covariates x_i. The average entropy of each method in each data set is shown in Table 2. The results demonstrate the fact that ARD has the largest variability on average in generating the relevance ranking.

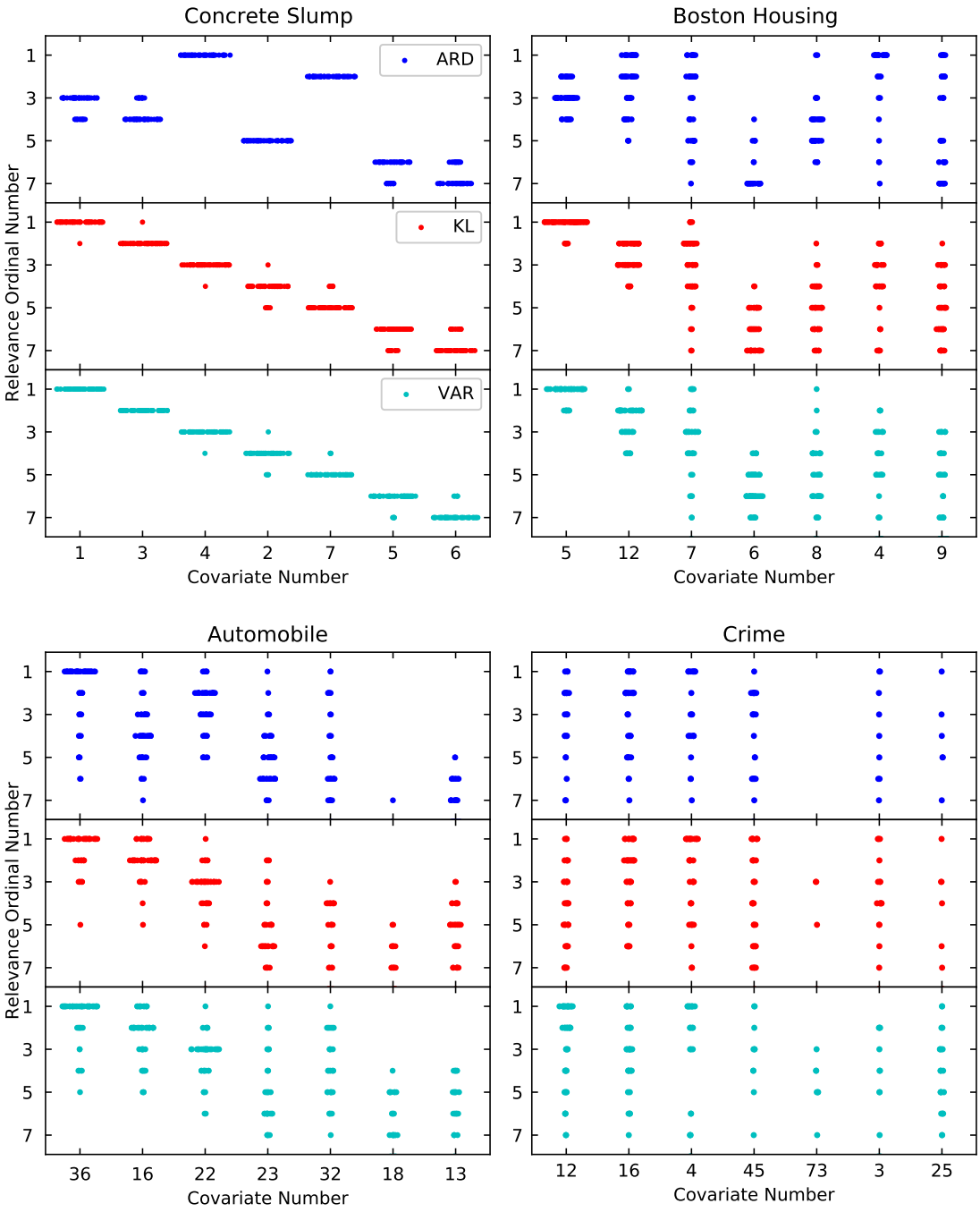

Figure 6: Scatter plots representing the frequency of relevance ordinal numbers between different training sets given to 7 covariates in the four data sets. The 7 covariates shown are selected based on the fact that they were the most relevant according to the VAR method. The markers in each ordinal number slot are jittered horizontally to visualize the number of points in each slot.

Table 2: Average entropy of each covariate choice.

Dataset	ARD	KL	VAR
Concrete Slump Test	7.3	5.6	3.1
Boston Housing	14.2	13.8	13.5
Automobile	11.6	10.7	10.5
Crime	6.1	5.7	5.9

To visualize the difference of the covariate preferences between ARD and the alternative methods, slices of the Gaussian process models fitted to the Boston Housing data are shown in Figure 7. The slices show the mean and standard deviations of the Gaussian process as a function of covariates 4 and 5, when the values of the other covariates are set to their respective means. In this training split, covariate 4 was picked first by ARD, and covariate 5 was picked first by KL and VAR. The plot illustrates that the latent function as a function of x_4 is relatively flat, but its length-scale is small due to the small oscillation. Conversely, as a function of x_5 it is quite rigid, but it has a larger slope and thus is more informative for predicting y. This is well in line with the observation made from the toy data set that ARD overly favours covariates with a nonlinear response.

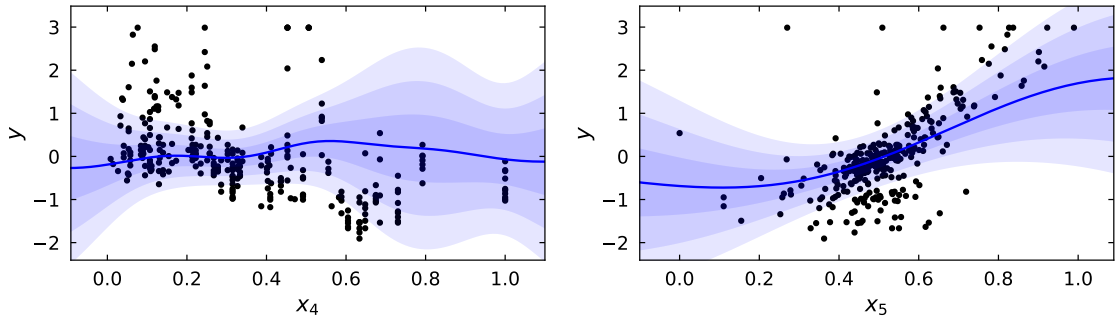

Figure 7: Slices of the Gaussian process model fitted to one training sample of the Boston Housing data set by maximizing the posterior of the hyperparameters. In this training sample, covariate x_4 (left) was considered the most relevant according to ARD. Covariate x_5 (right) was considered most relevant by the KL and VAR methods. The blue line is the mean, and the shaded areas represent one, two, and three standard deviations of the latent function.

7.3 Estimation of locally relevant covariates

In some cases, a covariate might have strong predictive relevance in some region of the input space, while having a small effect on the target variable on average. In some applications, the identification of such locally relevant covariates is important. Consider a hypothetical regression problem, where the covariates represent measurements to be made on a patient, and the dependent variable represents the progression of a disease. The information that some measurement has little relevance on average, but for some patients it is a clear indication of how far the disease has progressed, may provide essential information for medical professionals.

Because the KL and VAR methods compute relevance estimates at each training point, they produce a collection of relevance values in the regions of the input space where training points are located. The relevance values can then be addressed individually to assess the relevance of covariates in some subspace of the inputs. To demonstrate this, the KL relevance estimates were computed at each training point of the toy dataset with normally distributed inputs (56), and they are plotted together with the values of covariate x_8 in Figure 8. The red curve is a sketch of the underlying latent function $f_8(x_8)$ and is not scaled properly. While there is some noise, the distribution neatly captures both the flat and steep regions of the original sinusoidal function, exhibiting low and high relevance values, respectively. Because the toy data set is very simple to model, the uncertainties are close to equal everywhere, and the resulting KL relevance estimates are thus extremely correlated with the derivative of the latent function.

Figure 8 neatly captures one of the novel aspects of the KL and VAR methods. While automatic relevance determination outputs only a single number representing the nonlinearity of a covariate, the other methods give a collection of relevance values. A single nonlinearity value can represent a myriad of different functions, and while some of them can be relevant for the regression problem, others can be very irrelevant. The mean of the produced relevance collection can be used to assess the average relevance of a covariate, but the distribution can also be used to examine relevances locally. As shown in section 7.2, the averaging already provides an improvement to the assessment of predictive relevance, and the ability to consider the local relevances is a useful extra feature, improving the applicability of the methods.

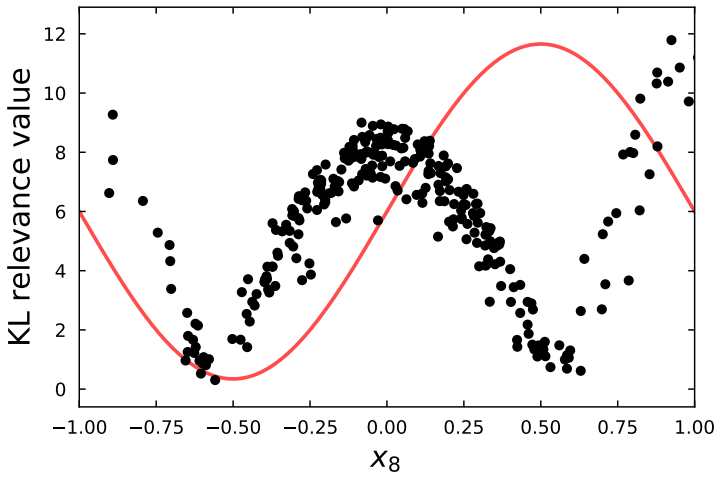

Figure 8: Pointwise KL relevance estimates together with the values of the eighth covariate, computed from a sample of 300 points from the toy dataset (56) with normally distributed input data. The red curve is a sketch of the original latent function $f_8(x_8)$ and is not scaled properly.

8 Summary and discussion

This thesis has studied Bayesian variable selection with the focus on Gaussian process models. The thesis has introduced two novel covariate selection methods, nicknamed KL and VAR, which were motivated theoretically and studied in a range of numerical experiments. The experiments included both simulated data sets and freely available benchmark data sets, and the new methods were thoroughly compared to automatic relevance determination, which is the prevalent covariate selection method in the Gaussian process literature.

The drawbacks of automatic relevance determination arise from the use of the length-scale parameters of the Gaussian process covariance function as an estimate for predictive relevance. Because the length-scale parameter primarily represents the nonlinearity of the latent function with respect to the corresponding covariate, ARD confuses nonlinear response with predictive relevance. Additionally, the weak identifiability of the length-scale parameters makes ARD less reliable and consistent as an estimator of relevance.

The experiments of this thesis showed that the newly introduced covariate selection methods yield improvements in several aspects compared to automatic relevance determination. First, both methods were shown to select submodels with increased predictive performance. The difference was most significant in small models with less than five covariates, but also larger models never performed worse than those selected by ARD. This is a result of the fact that ARD considers those covariates most relevant that have the most nonlinear response when the Gaussian process model is optimized by maximizing the posterior distribution of the hyperparameters. On the other hand, the alternative methods preferred covariates with a more linear but stronger response. The relevance estimates of these methods were highly dependent on the average partial derivative of the mean of the posterior Gaussian process, and this was shown to be a better indicator of predictive relevance.

Additionally, the experiments demonstrated that the KL and VAR methods were more consistent about the relevance ranking that they produced for the covariates when the training data was varied slightly. Between different splits to training and test data sets, ARD was shown to be the least consistent in all of the four benchmark data sets. The improved consistency makes the new methods more reliable for estimating the predictive relevance of covariates from limited data.

The new methods introduced in this thesis require computing relevance estimates for each covariate in each point of the training data, thus increasing the computational complexity compared to automatic relevance determination. The VAR method, in the form presented here, is also restricted to data sets where the dimensionality is

less than the number of data points. Additionally, the assumption about normally distributed inputs may cause errors in data sets that are far from Gaussian. Despite their limitations, the newly introduced covariate selection methods were shown to perform well in practice. More thorough experiments with different data sets and different types of models are still needed to further assess the usefulness of the methods.

References

Berger, J. O. (2013). *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley & Sons.

Betancourt, M. (2016). Identifying the optimal integration time in Hamiltonian Monte Carlo. *arXiv preprint, arXiv:1601.00225*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Cawley, G. C. and Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222.

Dupuis, J. A. and Robert, C. P. (2003). Variable selection in qualitative models via an entropic explanatory power. *Journal of Statistical Planning and Inference*, 111(1):77–94.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian Data Analysis*. Chapman & Hall, Third edition.

Gelman, A. and Pardoe, I. (2007). Average predictive comparisons for models with nonlinearity, interactions, and variance components. *Sociological Methodology*, 37(1):23–51.

George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230.

Goutis, C. and Robert, C. P. (1998). Model choice in generalised linear models: A Bayesian approach via Kullback-Leibler projections. *Biometrika*, 85(1):29–37.

GPy (since 2012). GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3:1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2008). *Feature Extraction: Foundations and Applications*. Springer.

Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM review*, 31(2):221–239.

Härdle, W. and Stoker, T. M. (1989). Investigating smooth multiple regression by the method of average derivatives. *Journal of the American Statistical Association*, 84(408):986–995.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401.

Hoffman, M. D. and Gelman, A. (2014). The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

Jefferys, W. H. and Berger, J. O. (1992). Ockham's razor and Bayesian analysis. *American Scientist*, 80(1):64–72.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine learning: proceedings of the eleventh international conference*, pages 121–129.

Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Prentice Hall.

Koller, D. and Sahami, M. (1996). Toward optimal feature selection. Technical report, Stanford InfoLab.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Lampinen, J. and Vehtari, A. (2001). Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml/index.php. Accessed: 2017-11-03.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006). Variable selection for Gaussian process models in computer experiments. *Technometrics*, 48(4):478–490.

MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, 4(3):415–447.

MacKay, D. J. (1994). Bayesian nonlinear modeling for the prediction competition. *ASHRAE transactions*, 100(2):1053–1062.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

Mohammed, R. O. and Cawley, G. C. (2017). Over-fitting in model selection with Gaussian process regression. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 192–205. Springer.

Neal, R. M. (1995). *Bayesian learning for neural networks*. PhD thesis, University of Toronto.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42.

Paananen, T., Piironen, J., Andersen, M. R., and Vehtari, A. (2017). Model selection for Gaussian processes utilizing sensitivity of posterior predictive distribution. *arXiv preprint, arXiv:1712.08048*.

Piironen, J. and Vehtari, A. (2016). Projection predictive model selection for Gaussian processes. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Piironen, J. and Vehtari, A. (2017). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735.

Raftery, A. E. and Zheng, Y. (2003). Discussion: Performance of Bayesian model averaging. *Journal of the American Statistical Association*, 98(464):931–938.

Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. In *Advances in neural information processing systems*, pages 294–300.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT press.

Redmond, M. and Baveja, A. (2002). A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382.

Riihimäki, J. and Vehtari, A. (2010). Gaussian processes with monotonicity information. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 645–652.

Savitsky, T., Vannucci, M., and Sha, N. (2011). Variable selection for nonparametric Gaussian process priors: Models and computational strategies. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 26(1):130.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:623–656.

Simpson, D., Rue, H., Riebler, A., Martins, T. G., and Sørbye, S. H. (2017). Penalising model component complexity: A principled, practical approach to constructing priors. *Statistical Science*, 32(1):1–28.

Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative observations in Gaussian process models of dynamic systems. In *Advances in neural information processing systems*, pages 1057–1064.

Stan Development Team (2017). *Stan Modeling Language Users Guide and Reference Manual*. http://mc-stan.org. Version 2.16.0.

Stein, M. L. (2012). *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

Stolzenberg, R. M. (1980). The measurement and decomposition of causal effects in nonlinear and nonadditive models. *Sociological methodology*, 11:459–488.

Titsias, M. K. and Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in neural information processing systems*, pages 2339–2347.

Vehtari, A. (2001). *Bayesian model assessment and selection using expected utilities*. PhD thesis, Helsinki University of Technology.

Vehtari, A. and Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228.

Yeh, I.-C. (2007). Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, 29(6):474–480.

Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5:1205–1224.

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261.

Appendix A

Rank one update of the Cholesky decomposition

This Appendix presents the method for obtaining the Cholesky decomposition of a submatrix with one row and one column removed. This is done by updating the Cholesky decomposition of the full matrix with a rank-one update (Hager, 1989). Denote the full matrix and its Cholesky decomposition as $\mathbf{\Sigma} = \mathbf{L}\mathbf{L}^\mathsf{T} \in \mathbb{R}^{p\times p}$. The goal is to obtain the Cholesky decomposition of the submatrix $\mathbf{\Sigma}_{-j,-j} = \mathbf{L}_{-j,-j}\mathbf{L}_{-j,-j}^\mathsf{T} \in \mathbb{R}^{(p-1)\times(p-1)}$, where the row j and column j are removed from the full matrix $\mathbf{\Sigma}$. A direct Cholesky decomposition of the submatrix has a computational complexity of $\mathcal{O}(p^3)$, but a rank one update has only $\mathcal{O}(p^2)$. If the parts of the lower triangular matrix $\mathbf{L}$ are denoted as

$$\mathbf{L} = \begin{array}{c} \\ <j \\ j \\ >j \end{array} \begin{array}{c} \begin{array}{ccc} <j & j & >j \end{array} \\ \begin{pmatrix} \mathbf{L}_A & \mathbf{0} & \mathbf{0} \\ \mathbf{l}_B^\mathsf{T} & l_{j,j} & \mathbf{0}^\mathsf{T} \\ \mathbf{L}_C & \mathbf{l}_D & \mathbf{L}_E \end{pmatrix} \end{array} \in \mathbb{R}^{p\times p}, \tag{58}$$

The corresponding triangular matrix of the submatrix $\mathbf{\Sigma}_{-j,-j}$ is obtained as

$$\begin{aligned} \mathbf{L}_{-j,-j} &= \begin{pmatrix} \mathbf{L}_A & \mathbf{0} \\ \mathbf{L}_C & \tilde{\mathbf{L}}_E \end{pmatrix} \in \mathbb{R}^{(p-1)\times(p-1)}, \\ \tilde{\mathbf{L}}_E\tilde{\mathbf{L}}_E^\mathsf{T} &= \mathbf{L}_E\mathbf{L}_E^\mathsf{T} + \mathbf{l}_D\mathbf{l}_D^\mathsf{T}. \end{aligned} \tag{59}$$

Because $\mathbf{l}_D$ is a vector, the modification to the Cholesky decomposition in equation (59) is a rank-one update.